



### **Science Arts & Métiers (SAM)**

is an open access repository that collects the work of Arts et Métiers Institute of Technology researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: <https://sam.ensam.eu>  
Handle ID: [.http://hdl.handle.net/10985/22984](http://hdl.handle.net/10985/22984)

#### **To cite this version :**

Ali SHAH GHAZANFAR, Arnaud POLETTE, Jean-Philippe PERNOT, Franca GIANNINI, Marina MONTI - Casebased tuning of a metaheuristic algorithm exploiting sensitivity analysis and design of experiments for reverse engineering applications - Engineering with Computer - 2022

Any correspondence concerning this service should be sent to the repository

Administrator : [scienceouverte@ensam.eu](mailto:scienceouverte@ensam.eu)





# Case-based tuning of a metaheuristic algorithm exploiting sensitivity analysis and design of experiments for reverse engineering applications

Ghazanfar Ali Shah<sup>1,2,3</sup> · Arnaud Polette<sup>1</sup> · Jean-Philippe Pernot<sup>1</sup> · Franca Giannini<sup>2</sup> · Marina Monti<sup>2</sup>

Received: 18 October 2021 / Accepted: 17 March 2022

© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

## Abstract

Due to its capacity to evolve in a large solution space, the Simulated Annealing (SA) algorithm has shown very promising results for the Reverse Engineering of editable CAD geometries including parametric 2D sketches, 3D CAD parts and assemblies. However, parameter setting is a key factor for its performance, but it is also awkward work. This paper addresses the way a SA-based Reverse Engineering technique can be enhanced by identifying its optimal default setting parameters for the fitting of CAD geometries to point clouds of digitized parts. The method integrates a sensitivity analysis to characterize the impact of the variations in the parameters of a CAD model on the evolution of the deviation between the CAD model itself and the point cloud to be fitted. The principles underpinning the adopted fitting algorithm are briefly recalled. A framework that uses design of experiments (DOEs) is introduced to identify and save in a database the best setting parameter values for given CAD models. This database is then exploited when considering the fitting of a new CAD model. Using similarity assessment, it is then possible to reuse the best setting parameter values of the most similar CAD model found in the database. The applied sensitivity analysis is described together with the comparison of the resulting sensitivity evolution curves with the changes in the CAD model parameters imposed by the SA algorithm. Possible improvements suggested by the analysis are implemented to enhance the efficiency of SA-based fitting. The overall approach is illustrated on the fitting of single mechanical parts but it can be directly extended to the fitting of parts' assemblies. It is particularly interesting in the context of the Industry 4.0 to update and maintain the coherence of the digital twins with respect to the evolution of the associated physical products and systems.

**Keywords** Sensitivity analysis · CAD model parameters · Simulated annealing · Reverse engineering · Digital twins

✉ Ghazanfar Ali Shah  
ghazanfar\_920@yahoo.com

Arnaud Polette  
arnaud.polette@ensam.eu

Jean-Philippe Pernot  
jean-philippe.pernot@ensam.eu

Franca Giannini  
franca.giannini@ge.imati.cnr.it

Marina Monti  
marina.monti@ge.imati.cnr.it

<sup>1</sup> LISPEN, EA 7515, HeSam, Arts et Métiers, 2 cours des Arts et Métiers, 13617 Aix-en-Provence, France

<sup>2</sup> IMATI, Consiglio Nazionale delle Ricerche, Via De Marini, 6, 16149 Genoa, Italy

<sup>3</sup> DIME-Dipartimento di Ingegneria meccanica, energetica, gestionale e dei trasporti, Università degli Studi di Genova, Genoa, Italy

## 1 Introduction

Today, being able to reconstruct or update 3D representations of existing products and systems has become mainstream to support the creation and exploitation of digital twins in the context of Industry 4.0 [1]. The relative ease of access to more or less sophisticated 3D acquisition devices has certainly accelerated the demand while extending the range of possible applications. However, point cloud processing and Reverse Engineering (RE) of CAD models are not yet fully automatized. The resulting models are not fully exploitable by CAD software and the patch-by-patch reconstructions do not meet the Industry 4.0 requirements [2]. Most of the existing tools for RE ends up with non-editable dead models in which the structure and shape of the CAD models are frozen without any possibility to act on their parameter values that are even

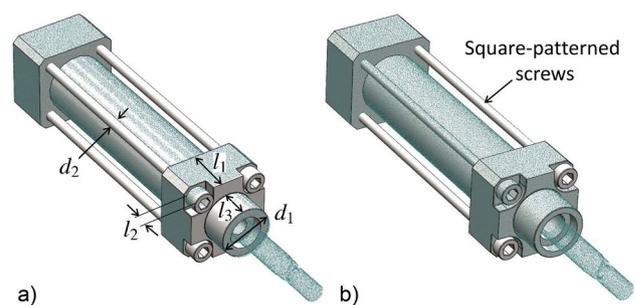
not accessible anyway. Only the position and orientation of the constitutive parts can be updated. Commercial software like CATIA V5, DesignX or SpaceClaim mostly rely on patch-by-patch reconstruction, with some capabilities to reconstruct a limited number of features like revolves, extrusions, etc. However, this possibility is limited when the part is complex with many features. The capabilities of dealing with the reconstruction of complete assemblies while satisfying constraints are hardly supported. The chosen references at the beginning of the reconstruction process normally go all along the way in the traditional RE and do not allow any modification at the end of the process, thus limiting the users to go back to any step during the reconstruction process. Even through a direct modeling strategy offered by some of those tools, the modification and update are still time-consuming as they act at a rather low-level, i.e., the patches. Some researchers are also working directly on the patches. For instance, Borhen Louhichi et al. [3] presented an approach to fit the B-Rep surfaces to the deformed meshes resulting from Finite Element Analysis (FEA). This approach allows for rebuilding the CAD model after analysis by extracting geometric information from the deformed mesh. Mesh partitions are paired with their underlying CAD topology prior to the FEA (Triangles to B-Rep Faces, segments to B-Rep edges). The approach is limited to basic primitive (i.e., sphere, plane and cylinder, etc.) to update the B-Rep by fitting primitive surfaces to the deformed mesh. Thus, current approaches fail to answer the update problem, and this is a major limitation which prevents the fast update of digital twins with respect to the evolution of the physical ones in the scope of the Industry 4.0. Moreover, not only they can be used to solve the update problem, but editable reverse engineered CAD geometries can also be modified in the later stages of the product development processes (PDP). In this paper, the update problem is transformed in a fitting problem for which the optimal parameter values of CAD models have to be found in order to minimize the deviation between the digital twins and the point clouds coming from the digitization of the physical twins.

There are few methods to reconstruct editable CAD geometries. Among them, metaheuristic algorithms like Simulated Annealing (SA), Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) have been cleverly used to optimize the parameters of the CAD geometries to fully fit in the point cloud obtained from data acquisition devices [4–7]. These metaheuristic algorithms try to find global optimal solutions by both diversifying and intensifying the search in a large solution space. Even if such algorithms proved to be very beneficial to support the RE process, the tuning of their setting parameters is not an easy task. This paper addresses the way SA-based reconstructions can be enhanced by identifying optimal default setting

parameters for the fitting of CAD geometries to point clouds of digitized parts.

The RE process described in [4] considers as inputs the digitized point cloud, the CAD model to be fitted and the set of its parameters to be modified to minimize the deviation between the two. The applied metaheuristic SA algorithm has a stochastic behavior [8], with few knowledge on how the CAD parameter values are to be changed throughout the optimization. Therefore, if the SA setting parameters are not properly chosen, the fitting process may end up with either under or over-fitting of the CAD parts. Figure 1a shows a bad fitting of a pneumatic pump consisting of multiple parts. Some control parameters of the CAD models that drive the shapes are under fitted (e.g.  $l_3$ ) while other are overfitted ( $l_1$ ,  $l_2$ ,  $d_1$  and  $d_2$ ). To avoid bad quality fitting of CAD models and assemblies, a tuned SA-based fitting algorithm is used with setting parameters obtained following the tuning strategy proposed in this paper (Fig. 1b).

It is well known that good parameter settings have a significant effect on the performance of an algorithm [9]. For example, SA algorithm is sensitive to the cooling factor, while a tabu search algorithm relies on a good choice of the tabu tenure [10]. A possible way to understand the influence between the output and parameters' evolution is sensitivity analysis [11, 12]. It has been used in various contexts, as, for example, for the parameter choices of SA for continuous network design [13], and various indices have been specified [14–16]. Robinson et al. [17] proposes an approach to determine which parameters defining the features in a CAD model need to be modified, and by what amount, to optimize component performance. It uses sensitivities computed for the parameters to determine the change required in each to optimize the component by first calculating the change of the objective function due to the perturbation of a CAD parameter. In this approach, design velocities refer to the rate of boundary surface movement due to a parameter perturbation, which is closely related to a shape fitting problem and the CAD parameter's sensitivity. The overall approach



**Fig. 1** Fitting of a CAD assembly on the point cloud of a digitized pneumatic pump: coarse result without tuning the parameters of the SA-based fitting algorithm (a); and good-quality result after tuning (b)

is unaffected by the topology changes and the complexity of the CAD model. The sensitivity analysis allows the evolution of the influence of the CAD model parameters to be monitored throughout the fitting process. Comparing the obtained sensitivity evolution curves to the changes of the parameters directly imposed by the SA algorithm provides hints to improve the fitting process. Some basic and advanced parameters are required by the SA-based fitting algorithm to reverse engineer CAD models and, in this paper, a new framework is proposed that couples sensitivity analysis of the CAD parameters to SA algorithm and identifies the default parameters configurations with the help of a design of experiments (DoEs). Within this framework, a database is created to store together the CAD models with their best SA parameters configuration. Later, if a new CAD template is to be reverse engineered following this SA-based fitting technique, a case-based approach can be applied for the setting of the SA parameters by reusing the values stored for the most similar CAD model in the database.

The contribution of this paper is threefold: (1) a new framework using design of experiments (DoEs) for the creation of a database of CAD models stored together with their best SA-based fitting configurations; (2) sensitivity analysis to study the influence of SA on the parameters of CAD models during the fitting process; (3) identification of the default setting parameters for SA-based fitting to obtained best-fitted geometries using DoEs. The proposed strategy for tuning the SA parameters is generic in the sense that it could be also applied to other problems where SA is needed, and for which a database of best configurations could be set up for case re-use through similarity check.

The paper is organized as follows. Section 2 reviews some works related to sensitivity analysis as well as to the way SA parameters can be tuned to improve its overall performance. SA-based fitting of CAD models is briefly discussed in Sect. 3. Section 4 details the overall framework for the identification of best setting parameters to be used by the adopted SA-based fitting algorithm. The results are discussed in Sect. 5 and the approach is validated on several examples including local and global fitting on real-scanned and virtually-generated point clouds. Lastly, Sect. 6 concludes the paper and discusses future works.

## 2 Related works

Studies and experimentations have shown that the SA setting parameters, e.g. initial temperature, initial starting position and maximum number of iterations have a clear effect on the final fitting quality. Many attempts have been made to optimize the SA algorithm by tuning its setting parameters. For instance, Yang Jin et al. [13] studied the effects of parameters choice on the performance of SA

algorithm to solve the continuous network design problem, but their findings and focus were mostly towards the temperature decrease rate and iteration number, without considering other parameters that also affect the quality of their SA-based optimization. Shi-hua Zhan et al. [18] proposed a new List-Based Simulated Annealing algorithm (LBSA) to solve the Traveling Salesman Problem (TSP) in which LBSA requires a novel list-based cooling schedule to control the decrease of temperature. In their method, the temperature list is adopted iteratively according to the topology of the solution space of the problem. The effectiveness and the sensitivity of the parameter of the list-based cooling schedule are illustrated on the TSP benchmark problem. In this method, a list of initial temperatures are generated and tested but again only temperature is considered to optimize the SA action towards the final solution.

Bellio et al. [19] used a single-stage SA optimization to solve the Curriculum-Based Course Timetabling (CB-CTT) problem. They proposed an extensive and statistically principled methodology for the parameters tuning procedure to model the relationships between the most relevant parameters of the solver and the features of the instance under consideration. The method tackles the parameters selection as a classification problem and builds a rule for choosing the set of parameters most likely to perform well for a given instance on the basis of given features. However, their study was limited to fewer instances to improve the previous results and out of 21 instances only 10 were improved. Some specific research has been devoted by Atiqullah et al. [20] to augment the convergence and related behavior of annealing algorithms by modifying its parameters, otherwise known as cooling schedule. They introduced an approach to tune the SA algorithm by combining algorithmic and parametric augmentations. They used a simpler cooling algorithm as compared to the complex schedules for designing a welded beam using their tuned up SA. Here again, a limited number of parameters is taken into account for the improvement of the SA algorithm, especially the cooling schedule.

In conclusion, it is obvious from the previous studies that SA-based optimization algorithms require some tuning of their setting parameters to get good-quality results in an efficient way. This is true for a wide range of applications, including the ones related to the RE of parametric CAD models. This paper presents a method to identify the best configuration of SA setting parameters when considering the optimization of CAD templates (e.g. parametric sketches, CAD parts and assemblies) to be fitted into point clouds of digitized parts. A framework is introduced to create a database of CAD parts stored together with their best SA parameters configuration. Then, when a new part is to be reconstructed using the SA-based fitting algorithm, the user can extract the best configuration settings from the available

database using similarity checks between the CAD model to be fitted and the ones available in the database.

### 3 Simulated annealing-based fitting of CAD models

Recently, following a part-by-part rather than a patch-by-patch strategy, a breakthrough has been achieved for the reverse engineering of editable CAD models fitting point clouds of digitized mechanical parts or assemblies [4]. The idea is to work directly at the level of the part whose parameters are modified by an optimization algorithm until a good-quality fitting is obtained. To address this, a metaheuristic algorithm was required to handle a large solution space.

#### 3.1 Simulated annealing algorithm

Simulated Annealing has been widely used in different applications to solve optimization problems. The algorithm was designed for a stochastic search problem that successfully avoids local minimum during the search process. It uses a probabilistic approach to move from one point to another in search of global optima. This transition process of finding the optimal solution in a large search space depends on the temperature and the change in the objective function. SA optimization algorithm mimics the physical process of heating a material and then slowly reducing its temperature that decreases the overall energy of the system (annealing) to remove the defects in the material. SA-based fitting of CAD models also works on the same principle where the overall energy (sum of the square of distances between the tessellated CAD part  $\mathcal{M}^p$  and the point cloud PC) is minimized as shown in Eq. (1). Other metaheuristics have been tested but have demonstrated a lower efficiency than SA. For instance, Particle Swarm Optimization (PSO) generates candidate solutions that can be significantly different from the initial position. This optimization strategy may result in configurations that are hard for the CAD modeler to update, and may even cause the software to crash.

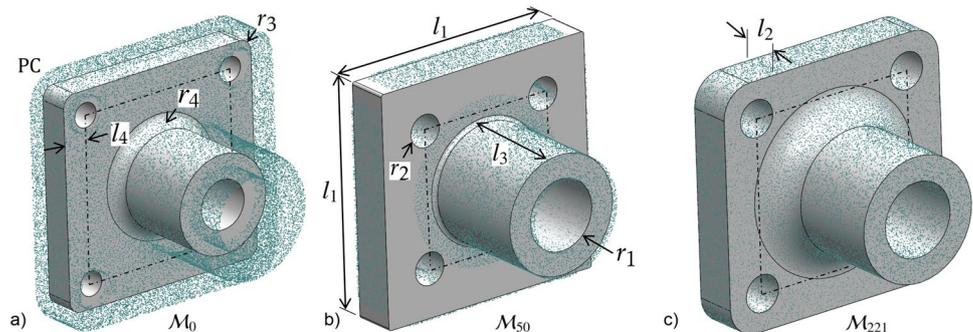
To better understand the fitting technique, the method is exemplified on the fitting of a rectangular flange as shown in Fig. 2. The input is the point cloud PC to which a parameterized CAD model  $\mathcal{M}_0$  is to be fitted. The point cloud is composed of  $N$  points  $PC[i]$ , with  $i \in [1..N]$ , and the CAD model is parameterized by  $N_p$  control parameters  $p_k$ , with  $k \in [1..N_p]$ . Here, there are 8 control parameters overall: the lengths  $l_1$  to  $l_4$ , and the radii  $r_1$  to  $r_4$ . During the prearrangement step, the user locates in a coarse manner the part inside the point cloud, and chooses the control parameters of the CAD part to be optimized by the SA algorithm, this initializes the parameters values  $p_{k,0}$  (Fig. 2a). At each iteration  $j$ , the SA algorithm then tries to modify the parameters values  $p_{k,j}$  until the updated CAD model perfectly fits the point cloud (Fig. 2b, c). The quality of the fitting is assessed thanks to an energy function that characterizes the overall deviation between the point cloud and the CAD model. The process stops when the energy function no longer decreases with respect to a given accuracy  $\varepsilon_s$ , or when a max number of iterations  $M_{iter}$  is reached.

At an iteration  $j \geq 0$  of the SA algorithm, the energy function to be minimized is evaluated as follows:

$$E(p_{1,j}, \dots, p_{N_p,j}) = \sum_{i=1}^N d^2(PC[i], \mathcal{M}_j^p(p_{1,j}, \dots, p_{N_p,j})) \quad (1)$$

wherein  $\mathcal{M}_j^p$  represents the tessellation of the CAD model  $\mathcal{M}_j$  after its update by the CAD modeler at the  $j$ -th iteration of the SA algorithm, and  $d(\text{point}, \text{mesh})$  is the distance function that returns the closest distance between a *point* and a *mesh*. Distance computation is performed using CloudCompare called in batch mode to compute the nearest triangle distance against each point in the point cloud. Alternatively, point-to-point distance can also be used as an energy function if the tessellated mesh is sampled with points [5]. As detailed in [4], this process is further improved while allowing points of the PC to be filtered step after step, so as to allow local fitting of a part in the point cloud of a digitized mechanical assembly.

**Fig. 2** Global fitting of a rectangular flange defined by 8 control parameters: **a** initial configuration after the prearrangement step, **b** fitted part after 50 iterations of the SA algorithm, **c** final fitted part minimizing the deviation with the point cloud ( $\varepsilon_s=10^{-1}$  and  $M_{iter}=1000$ )



### 3.2 Setting parameters for SA algorithm

The adopted SA algorithm is controlled by certain setting parameters to allow the fitting of a CAD template in a given point cloud. The selection of the right setting parameters plays an important role to achieve a high-quality solution and it remains the crucial step that directly affects the final fitting results. Setting parameters are classified into two groups: the mandatory parameters and the advanced ones, which all together define a so-called parameters configuration  $C_k$ . For instance, the parameter values used for the SA algorithm to reconstruct the rectangular flange of Fig. 2 are shown in Table 1. Each parameter is detailed in the next sections. On one hand, the mandatory parameters are the essential parameters of SA and they must be specified to initiate the optimization. The final fitting results are highly influenced by their values, and they may produce unstable results if not selected wisely. On the other hand, additional parameters are introduced and result in a set of advanced tuning options, namely the weighting and grouping strategies. Each of them can be tuned differently depending on the fitting problem, and this can be tricky. Thus, in this paper, a methodology is developed to support the tuning of those parameters and the identification of a proper configuration  $C_k$ . The idea is to build a database of best configurations known to produce quality results on several fitting examples, and then make use of a similarity assessment approach to find out the most similar fitting configuration with respect to a new CAD model to be fitted.

### 3.3 Description of the setting parameters

The setting parameters are briefly explained below, whereas the values they can take are presented when detailing the Design of Experiments that has been set up.

Regarding the mandatory parameters, annealing function (*AnnealingFcn*) is used by SA to generate a random trial point for the next point. The algorithm chooses the

distance of the trial point from the current point by a probability distribution with a scale depending on the current temperature. Temperature function (*TemperatureFcn*) is used to update the temperature. The algorithm systematically lowers the temperature, storing the best point found so far. Maximum stall iterations (*StallLimit*) is used as one of the stopping criteria. SA keeps track of the average change in the function value for *StallLimit* iterations and the optimization process stops if the SA does not find any better solution point within the *StallLimit*. Function tolerance (*FuncToler*) is another stopping criterion of SA according to which algorithm runs until the average change in the value of the objective function is less than the value of *FuncToler*. Initial temperature (*InitialTemp*) controls the overall search results for the next candidates of parameters  $p_k$ . For each dimension, the temperature is used to limit the extent of search in that dimension. Finally, the *InitialPosition* of the CAD model also has influence on the final result as SA chooses the distance of the trial points from the initial values of the parameters  $p_k$  by a probability with a scale depending on the initial temperature.

As for the advanced parameters, a weighting strategy for the parameters  $p_k$  can be used to avoid under and over-fitting of small features like fillets and chamfers which in some cases are neglected by the SA algorithm due to a small contribution in the overall energy (see Sect. 4.5). In addition, a grouping strategy of parameters  $p_k$  can be selected as another mean of avoiding under and over-fitting of small features especially when there are many parameters controlling the CAD geometry. Grouping of parameters  $p_k$  can be done according to their sensitivities (Sect. 4.3) or according to the type of features they refer to (Sect. 4.4). For the cases where the number of parameters is small, one single group can be considered.

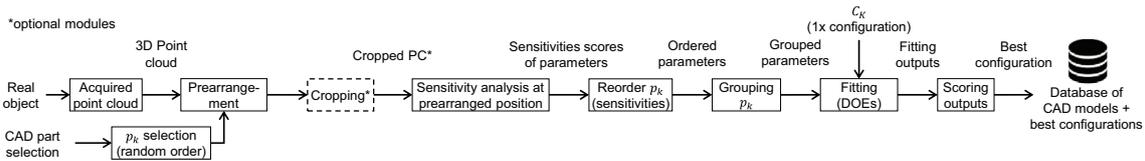
## 4 Framework for the selection of default configuration for SA-based fitting

This section introduces the framework developed to support the identification of the best configuration  $C_k$  for a given CAD model to be fitted in a point cloud. The idea is to first build a database of fitting examples storing both the CAD models together with their associated best parameter configurations. Then, for a new CAD model to be fitted, it is possible to refer to the database by means of a similarity assessment [21], to find out and then use the best configuration associated to the most similar fitting example.

Thus, the framework can be decomposed in two parts: the one used for the generation of the database (Fig. 3), and the one used to look for a similar configuration when a new CAD model is to be fitted (Fig. 9). This two-part framework is generic and could be used in other contexts. To create

**Table 1** Values of the setting parameters split in two categories and gathered together in a so-called configuration  $C_k$  used for fitting the rectangular flange

Categories	Parameters	Values
Mandatory	AnnealingFcn	annealingfast
	TemperatureFcn	temperatureexp
	StallLimit	25
	FunctionToler	1.00E-01
	InitialTemp	10
	InitialPosition	Position-02
Advanced	Weighting $p_k$	NoWeight
	Grouping $p_k$	NoGroup



**Fig. 3** Overall framework used to populate a database of CAD models stored together with their best configurations  $C_k$  for the SA

such a database, CAD models are fitted using SA annealing algorithm while playing with each setting parameter to get a good quality fitting and identify the associated best parameters configuration. Various parts are globally and locally fitted to their respective point clouds, obtained either from a laser scanner or virtually generated to get as-scanned point clouds [22]). Many possible configurations of the setting parameters for the SA algorithm are tested using DoEs to identify and store the best configuration in the database of CAD models. For simplicity, the first part of the framework has been explained on the fitting of a rectangular flange (Fig. 2). Thus, Sects. 4.1–4.7 explain how to identify best configurations and populate the database, whereas Sect. 4.8 details how to use the database, in a second step and for an unknown CAD model to be fitted, to retrieve the best configuration associated to the most similar fitting example.

**4.1 Selection, prearrangement and cropping**

Following the framework of Fig. 3, the process of adding an instance to the database starts with the selection of the initial parameterized CAD model  $M_0$  to fit in a given point cloud PC obtained after data acquisition. The models to be fitted can come from an existing database available in the company when considering for instance the need to either update the DMU of a system/product or control its assembly. They can also be roughly and rapidly sketched and parameterized starting from scratch using the CAD modeler. Here, it is assumed that the CAD models are parameterized at the level of the parts (e.g., lengths, diameters and angles parameterizing the features) and/or at the level of the assemblies (e.g., lengths and angles parameterizing the relative positions and orientation of the parts the ones with the others). These are user-specified parameters that have a functional meaning. A preprocessing step could be performed to decrease the interdependency between the parameters, keeping in mind not to break any design rule. In random order, all the parameters  $p_k$  controlling the shape of the CAD model are selected for the optimization. Next, the CAD model is manually prearranged with the given point cloud PC (Fig. 2a) as close as possible. This is an important step as this method is not robust if the prearranged CAD model is not in the close vicinity of the final solution. The point cloud can optionally be segmented if the intended optimization case is for the local fitting. This happens

when treating assemblies, where the template corresponds to a sub-part of the acquired object and thus only a segment of the point cloud take part in the optimization process, otherwise, the whole point cloud is utilized for the global fitting as required when the template geometry corresponds to the whole object acquired.

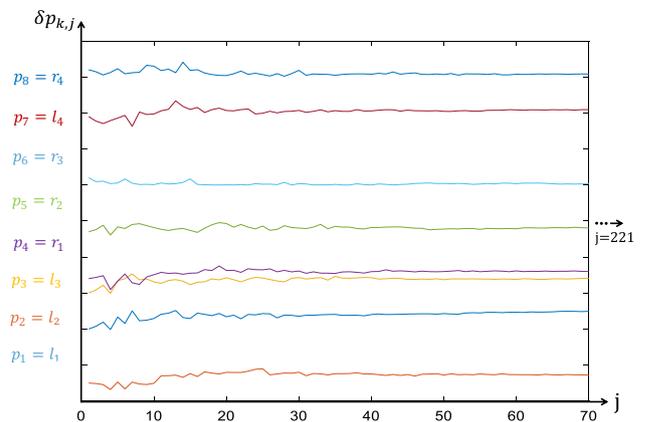
**4.2 Evolution of parameters  $p_k$  during SA-based fitting**

In order to understand how SA algorithm modifies the values of the parameters step after step, the evolution of the parameters is tracked throughout the fitting process. This can be done either in an absolute or relative manner, as follows:

$$\delta p_{k,j} = p_{k,j+1} - p_{k,j} \tag{2}$$

$$\Delta p_{k,j} = \frac{\delta p_{k,j}}{p_{k,j}} \tag{3}$$

Thus, the value  $\delta p_{k,j}$  characterizes the absolute evolution of the parameter  $p_k$  between iterations  $j$  and  $j + 1$ , whereas  $\Delta p_{k,j}$  refers to its relative evolution. At each iteration  $j$  of the SA algorithm, it is therefore possible to study the evolution of each parameter  $p_k$ , and see the decision taken by the SA algorithm on how to modify the parameter value for the next step  $j + 1$ . This is illustrated in Fig. 4 for the fitting of



**Fig. 4** Changes of  $\delta p_{k,j}$  imposed by the SA algorithm at each iteration  $j$  of the fitting process on the rectangular flange

the rectangular flange and using setting parameters for the SA listed in Table 1. From these evolution curves, it can be seen that all the parameters are modified more or less in the same manner.

Following the configuration  $C_\kappa$  of Table 1, the final absolute  $\delta p_k^F$  and relative deviations  $\Delta p_k^F$  of the parameters are computed and the results are listed in Table 2:

$$\Delta p_k^F = \frac{|\delta p_k^F|}{|p_k^D|} = \frac{|p_k^D - p_k^F|}{|p_k^D|} \quad (4)$$

where  $p_k^F$  are the final values of the parameters controlling the fitted CAD model, and  $p_k^D$  are the original values of the ground truth CAD model that has been used to virtually generate the given point cloud [4]. The deviations of some parameters are higher, e.g., the two radii  $r_3$  and  $r_4$ . This is due to the fact that SA neither takes into account the type of parameters nor how much they contribute to the overall energy of Eq. (1). To tackle this problem a sensitivity analysis of parameters is required, which is detailed in the next section.

### 4.3 Sensitivity analysis

The SA algorithm used for fitting of CAD models works as a black box and neither takes into account the semantics associated to the various parameters  $p_k$  nor the way their variations may influence the evolution of the energy function step after step. Indeed, while acting on the variables to be optimized, the SA algorithm does not distinguish for instance parameters like length, width, thickness and radius that characterize the features, and all the variables are considered in the same way. It also does not take into account the fact that the influence of a parameter variation on the energy may vary a lot depending on whether the parameter is associated to a structural feature (e.g., the length  $l_1$  of the rectangular flange in Fig. 2b) or to a detail feature (e.g., the radius  $r_4$ ).

To better compare the influence of the parameters, a sensitivity estimator  $S_{k,j}$  is here introduced to compute at

each step  $j$  of the optimization process the sensitivity of the energy function with respect to each parameter  $p_k$ :

$$S_{k,j} = \frac{\partial E}{\partial p_k} |_{(p_{1,j}, \dots, p_{N_{p_j},j})} \quad \text{with } j \geq 0 \quad (5)$$

Since the energy function is computed using the nearest point algorithm, it cannot be differentiated. Thus, the sensitivities are approximated using a first-order finite difference scheme (either forward or backward) directly controlled by the step  $h_k$  chosen for the corresponding parameter  $p_k$ :

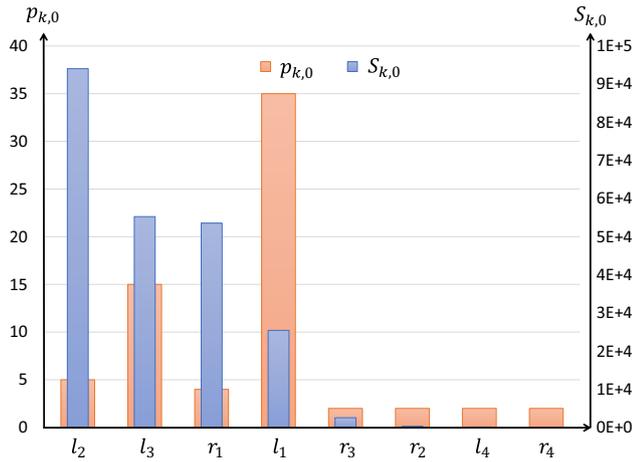
$$S_{k,j}^+ = \frac{E(p_{1,j}, \dots, p_{k,j} + h_k, \dots, p_{N_{p_j},j}) - E(p_{1,j}, \dots, p_{k,j}, \dots, p_{N_{p_j},j})}{h_k} + O(h_k), \quad \text{with } j \geq 0 \quad (6)$$

$$S_{k,j}^- = \frac{E(p_{1,j}, \dots, p_{k,j}, \dots, p_{N_{p_j},j}) - E(p_{1,j}, \dots, p_{k,j} - h_k, \dots, p_{N_{p_j},j})}{h_k} + O(h_k), \quad \text{with } j \geq 0 \quad (7)$$

Here, the term  $O(h_k)$  represents the order of magnitude of the approximation. More specifically, performing such a sensitivity analysis using all the parameters of the CAD model at the prearranged position ( $j = 0$ ) returns the sensitivities  $S_{k,0}$  of each parameter. This shows how much each parameter  $p_k$  affects the change of the overall energy. For example, the sensitivity analysis of the rectangular flange having 8 control parameters is performed at its prearranged position (Fig. 2a) and the resulting forward sensitivity against each parameter  $p_k$  is shown in Fig. 5. Thus, parameters  $l_2, l_3, r_1$  and  $l_1$  are the ones that modify the global energy  $E$  the most, while the influence of  $r_2, l_4$  and  $r_4$  is almost negligible. It can also be seen that sensitivities are independent of the magnitude of the parameter values  $p_{k,0}$  at prearranged position. After the computation of the sensitivities, parameters are rearranged in a decreasing order according to their sensitivity level (Fig. 5), and this helps grouping the parameters as discussed in Sect. 4.4.

**Table 2** Results for the global fitting of rectangular flange following the configuration  $C_\kappa$  of Table 1

Groups	$p_k$	$p_k^0$ (mm)	$p_k^D$ (mm)	$p_k^F$ (mm)	$\delta p_k^F$ (mm)	$\Delta p_k^F$
$G_1$	$l_2$	5	7	6.8313	0.17	0.0241
	$l_3$	15	20	19.975	0.02	0.0012
	$r_1$	4	6	6.0901	-0.09	0.0150
	$l_1$	35	40	39.719	0.28	0.0070
	$r_3$	2	5	3.4563	1.54	0.3087
	$r_2$	2	3	3.1657	-0.17	0.0552
	$l_4$	2	3	3.0214	-0.02	0.0071
	$r_4$	2	5	3.6982	1.30	0.2604



**Fig. 5** Sensitivities  $S_{k,0}$  of the 8 parameters  $p_k$  of the rectangular flange at prearranged position ( $j = 0$ )

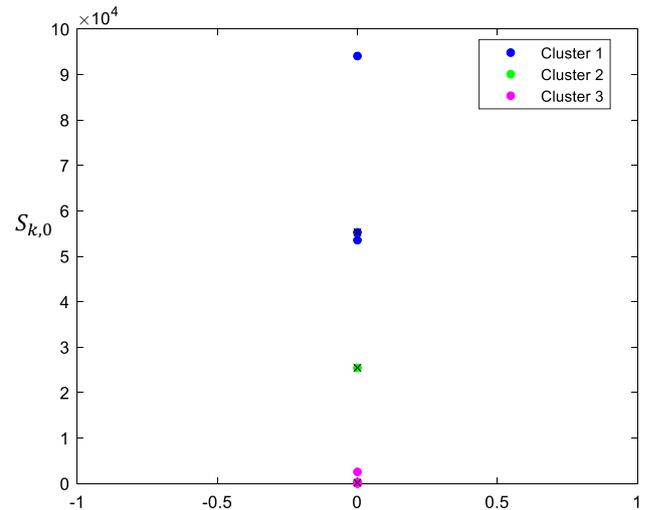
#### 4.4 Grouping strategies for parameters $p_k$

As part of the advanced SA settings, parameters  $p_k$  can optionally be clustered in several groups. Actually, grouping the parameters allows for a better and more stable fitting, and it also reduces the risk of getting the optimization process stuck in a local minimum. Conversely, considering all parameters in a single group may result in under or overfitting of small features. Thus, three grouping strategies are proposed:

- *Grouping according to sensitivity level* can be performed in an automatic manner using the K-means clustering technique. In this case, the sensitivity values are directly used as input of the K-means clustering algorithm that partitions the points into a number  $K$  of clusters. The resulting partitions minimize the sum, over all clusters, of the within-cluster sums of point-to-cluster-centroid distances. By default, K-means uses squared Euclidean distances. For the rectangular flange, this grouping strategy then returns three groups as shown in Fig. 6. The max value for the number  $K$  of clusters is assumed as:

$$K = \min \{3; N_p\} \quad (8)$$

- *Grouping according to the level of details* of the shape can be performed in a semi-automatic manner using a three-level organization, and following the commonly adopted CAD modeling strategies:  $G_1$  groups the parameters of the assembly structure as well as the ones driving the structural features (e.g. pockets, revolutions),  $G_2$  gathers the parameters of the detail features



**Fig. 6** Grouping parameters at prearranged position ( $j = 0$ ) according to sensitivity levels and using K-means clustering technique ( $K = 3$ )

(e.g. holes, ribs) and  $G_3$  includes the parameters used to finalize the shapes (e.g. fillets, chamfers).

- *NoGroup* strategy can be adopted by considering all the parameters in a single group.

#### 4.5 Weighting strategies

As part of the advanced SA settings, different weighting strategies can be followed to allow heterogeneous scaling on the parameter values as the SA algorithm goes on. Indeed, the trial points generated by *annealingFcn* basically apply the same scale of distribution of distances on all parameters  $p_k$ , independently of the type of feature they refer to. This is considered as a *NoWeight* weighting strategy. As a consequence, in some cases, due to small contributions in the overall energy, the energies related to small features like fillets and chamfers are neglected by the SA algorithm and this may cause under and overfitting of such features. To avoid this, weights can be put on the parameters at each iteration  $j$  of the fitting process. Thus, after performing the sensitivity analysis at the prearranged position (Sect. 4.3), the parameters are rearranged according to the decreasing order of their sensitivity levels (first parameter being the most sensitive and the last as the least one). Then, the idea is to cleverly change the scale distribution of *annealingFcn* that is generally uniform with equal weightage for all the parameters. To do so, two strategies are here proposed. To give more weightage to the least sensitive parameters, *LinearWeights* can be applied to all the parameters. The first parameter being the highest sensitive is given a weightage value of 1 (no weight) and for the trailing parameters 5% increment is added (i.e. 1.05 weight for the second parameter

and 1.10 for the third and so on). In this way, the last parameter (least sensitive) has the highest weight, which helps the evolution of the parameter not to remain blocked at the local minimum. Alternatively, individual weights or *SelectedWeights* can be provided to the least sensitive parameters. In the proposed method, *SelectedWeights* are only given for fillets and chamfers, which weight 50% more than the other parameters to avoid local minimum.

#### 4.6 Design of experiments set up

A configuration of settings  $C_k$  is composed of 8 setting parameters (6 mandatory and 2 advanced), each of which influencing the final quality of fitting. Thus, for a given CAD model to be fitted, testing all possible combinations of those parameters would result in a significantly large number of experiments to be conducted. To overcome this issue, a limited number of experiments is conducted by setting up a Design of Experiments (DoEs) that focuses on a restricted set of configurations  $C_k$ , among which the best configuration is to be identified for each sample of the database. Taguchi orthogonal array technique has been used to conduct a minimum number of experiments [23]. All the setting parameters with three possible levels are listed in Table 3. One of the parameter, namely *AnnealingFcn*, has 2 levels ( $2^1$ ) whereas the others have 3 levels ( $3^7$ ). According to the number of parameters and the number of levels, the  $L_{18}(2^1, 3^7)$  table of Taguchi orthogonal array has been chosen [24]. During each batch of 18 experiments, the quality of the fitting is observed on the basis of three criteria introduced in the next subsection.

More precisely, to help SA in setting the trial points distance distribution, *AnnealingFcn* can rely on two options, either *annealingfast* or *annealingboltz* [25, 26]. *Annealingfast* function generates a point equal to the current point plus a random step of size equal to the current temperature. While *annealingboltz* function generates a point based on the current

point and the current temperature using a multivariate normal distribution. The *TemperatureFcn* can be customized with three options: *temperatureexp*, *temperaturefast* or *temperatureboltz* [25, 26]. In the *temperatureexp* scheme, the temperature at any given step is 0.95 times the temperature at the previous step. This causes the temperature to go down slowly at first but ultimately get cooler faster than other schemes. If another scheme is desired, e.g., *boltzman* schedule or *fast* schedule annealing, then *temperatureboltz* or *temperaturefast* can be used respectively. For the termination criteria, *StallLimit* is provided with three values of 10, 25, and 50 iterations. For all three values, the SA algorithm will be tested if it can find the next better solution of parameters  $p_k$  from the current position, within given limits without spending too many iterations. For instance, 10 iterations will provide a small window for the SA to find the next best solution, while 50 iterations will provide sufficient time but it also increases the overall optimization time. *FunctionToler* is a second stopping criterion that can also be customized with three input values as shown in Table 3. The values are closed to zero to force the algorithm to find new best values of parameters  $p_k$  until the overall energy returned by the objective function is equal to one of these three values.

The selection of the initial temperature  $T_0$  follows the method suggested by [4], i.e. while looking for the optimal initial temperature within a given window of temperatures, and the one which provides the least energy is chosen. For all the CAD models added in the database, 10 is identified as the best initial temperature, and to conduct DoEs, two more values are chosen around the best-identified temperature by putting a  $5^\circ$  change on the positive and negative sides to test different temperatures. Hence, three initial temperatures are chosen for the DoEs, i.e., 5, 10, and 15. Analogously, for the initial position, three initial values are selected for the DoEs. All three initial positions are chosen by roughly positioning the CAD model close to the point cloud and by modifying the control parameters  $p_k$  of the rectangular flange to have a percentage volume difference of roughly 20%, 15% and 10% for Position-01, Position-02 and Position-03 respectively. This also initializes the values of the control parameters of the CAD models at initial position  $p_k^0$ , values that are then used for the initialization of the optimization process. Because of the random nature of the SA, the results obtained from DoEs can be biased. Indeed, a SA run with a certain set of parameters can produce exceptional results compared to what it would produce most of the time. In order to smooth this effect, for each DoE, SA is run three times and the mean of the three output measurements is finally computed.

#### 4.7 Scoring DoEs results for the best configuration selection

To identify the best possible configuration for the SA-based fitting algorithm, a scoring approach is used to rank the 18

**Table 3** List of setting parameters with corresponding three levels ( $level_{1-3}$ ) used for setting up the DoEs

Parameters	level <sub>1</sub>	level <sub>2</sub>	level <sub>3</sub>
AnnealingFcn	annealingboltz	annealingfast	–
TemperatureFcn	temperatureexp	temperatureboltz	temperaturefast
StallLimit	10	25	50
FunctionToler	0e+00	1e-01	1e-06
InitialTemp	5	10	15
InitialPosition	Position-01	Position-02	Position-03
Weighting $p_k$	LinearWeights	SelectedWeights	NoWeight
Grouping $p_k$	Group (Sensitivity)	Group (Features)	NoGroup

configurations involved in the DoEs. Table 4 shows all the configurations and their corresponding input values used for the SA-based optimization. Each of the 18 configurations  $C_k$  is evaluated on the basis of three criteria: (1) the number of iterations  $j_{max}$  taken by SA to find the optimal solution, (2) the final energy  $E_{j=j_{max}}$  obtained at the end of the optimization, (3) the sum of the final absolute deviations  $\Delta p_{k,j_{max}}$  of all parameters. Figure 7 shows the results obtained when running the 18 configurations on the rectangular flange example. For sake of clarity, the results for the three criteria are normalized and represented as a percentage using  $C_1$  as a reference. Thus, for  $C_1$ , each criterion is evaluated as 100% and the rest of the configurations are compared to it for normalization. It can be clearly seen that each configuration  $C_k$  ( $k \in [1..18]$ ) has different outcomes. For example, the result of  $C_7$  shows a smaller sum of absolute deviations, with less number of iterations but with a higher final energy. While

$C_{17}$  shows opposite results where it took more iterations to complete the fitting, with better results in terms of absolute deviations and final energy.

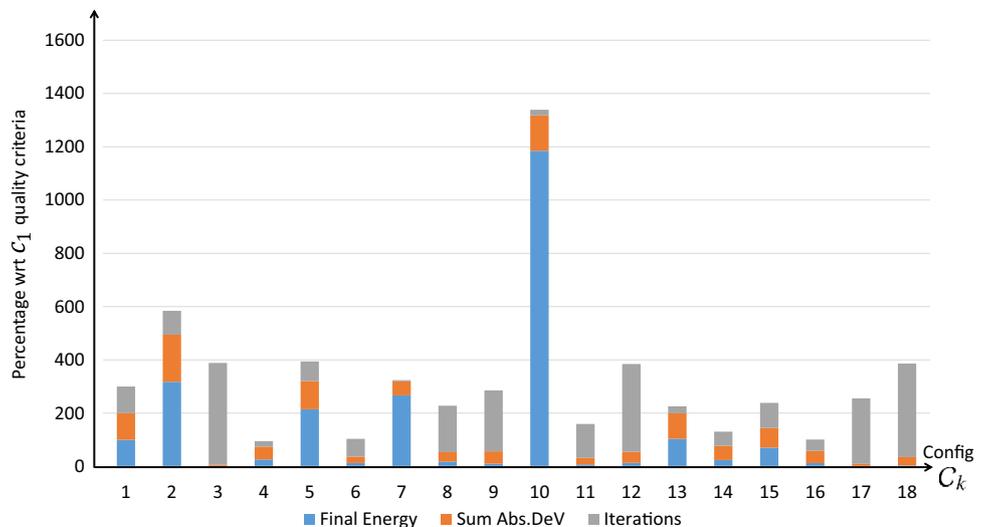
Then, to identify the best configuration among the 18 for a given fitting example, the three quality criteria, i.e. *Final-Energy*, *Sum Abs.Dev* and *Iterations* are ranked in an ascending order starting from the lowest to the highest value. To compute the final score of a configuration, the independent ranks of the three quality criteria are then sum up, and the configuration depicting the minimum score is considered as best configuration for this fitting example. Figure 8 shows the scores related to each configurations of the DoEs run on the rectangular flange.

For the rectangular flange, configuration  $C_6$  appears to be the best configuration as it shows the lowest score of 12 with its setting parameters highlighted in green in Table 4. In this particular case, *annealingboltz* with *temperatureboltz*

**Table 4** Fitting of rectangular flange for 18 configurations  $C_k$  ( $k \in [1..18]$ ) of DoEs using  $L_{18}$  table of Taguchi orthogonal array

$C_k$	AnnealingFcn	TemperatureFcn	StallLimit	FunctionTolerance	Grouping $p_k$	Weighting $p_k$	InitialTemp	InitialPosition	Scores
1	annealingboltz	temperatureexp	10	0.00E+00	NoGroup	NoWeight	5	Position-01	25
2	annealingboltz	temperatureexp	25	1.00E-01	Group(Features)	LinearWeights	10	Position-02	21
3	annealingboltz	temperatureexp	50	1.00E-06	Group(Sensitivity)	SelectedWeights	15	Position-03	19
4	annealingboltz	temperatureboltz	10	0.00E+00	Group(Features)	LinearWeights	15	Position-03	16
5	annealingboltz	temperatureboltz	25	1.00E-01	Group(Sensitivity)	SelectedWeights	5	Position-01	25
6	annealingboltz	temperatureboltz	50	1.00E-06	NoGroup	NoWeight	10	Position-02	12
7	annealingboltz	temperaturefast	10	1.00E-01	NoGroup	SelectedWeights	10	Position-03	15
8	annealingboltz	temperaturefast	25	1.00E-06	Group(Features)	NoWeight	15	Position-01	16
9	annealingboltz	temperaturefast	50	0.00E+00	Group(Sensitivity)	LinearWeights	5	Position-02	18
10	annealingfast	temperatureexp	10	1.00E-06	Group(Sensitivity)	LinearWeights	10	Position-01	25
11	annealingfast	temperatureexp	25	0.00E+00	NoGroup	SelectedWeights	15	Position-02	22
12	annealingfast	temperatureexp	50	1.00E-01	Group(Features)	NoWeight	5	Position-03	16
13	annealingfast	temperatureboltz	10	1.00E-01	Group(Sensitivity)	NoWeight	15	Position-02	23
14	annealingfast	temperatureboltz	25	1.00E-06	NoGroup	LinearWeights	5	Position-03	18
15	annealingfast	temperatureboltz	50	0.00E+00	Group(Features)	SelectedWeights	10	Position-01	24
16	annealingfast	temperaturefast	10	1.00E-06	Group(Features)	SelectedWeights	5	Position-02	18
17	annealingfast	temperaturefast	25	0.00E+00	Group(Sensitivity)	NoWeight	10	Position-03	13
18	annealingfast	temperaturefast	50	1.00E-01	NoGroup	LinearWeights	15	Position-01	15

**Fig. 7** Results of the DoEs for the rectangular flange normalized taking  $C_1$  as a reference



for SA annealing produced the best result of the fitting. The *StallLimit* is 50 which means the algorithm will have about 50 iterations to look for new best values of parameters, otherwise the algorithm will terminate. *StallLimit* with 50 iterations will give sufficient time to the algorithm to search for the next best set of parameters  $p_k$ . Furthermore, as this example is rather simple, *NoWeight* and *NoGroup* are needed, and Position-02 is sufficient to start the optimization process.

Finally, once this framework defined, a database can be created and populated while storing CAD models together with their best configurations. Hence, as discussed in the next subsection, the values stored in the database can be used as references to configure the SA setting parameters when considering a new CAD model to be fitted in a point cloud using the adopted SA-based fitting technique.

### 4.8 Fitting new CAD template reusing an acquired knowledge

As introduced at the beginning of Sect. 4, once created a database gathering CAD models together with their best

configurations for point cloud fitting, if the user wants to reconstruct a new CAD model, the second framework illustrated in Fig. 9 should be followed. The idea is to make use of a similarity assessment tool to identify the CAD model of the database which resembles the most to the new fitting case, and then use its best configuration to tune the SA setting parameters. The most suitable tool for this work would be a method capable of retrieving parts with similar shapes and similar parameterization. However, not being aware of the availability of such a tool and being outside the scope of this work to define new methods for similarity assessment, the tool of Giannini et al. [21] has been used here to assess the similarity. As terms of results, this tool returns a similarity matrix that helps in the identification of the CAD model that has the closest geometric description with the new CAD geometry to be fitted. Once the most similar geometry identified from the database, the associated best configuration can be used for the new fitting. The user still needs to specify which control parameters  $p_k$  of the new CAD model should be considered as variables of the optimization process. At the end, using this best configuration extracted from the database, the SA-based fitting algorithm

Fig. 8 Scores of DoEs result for the rectangular flange

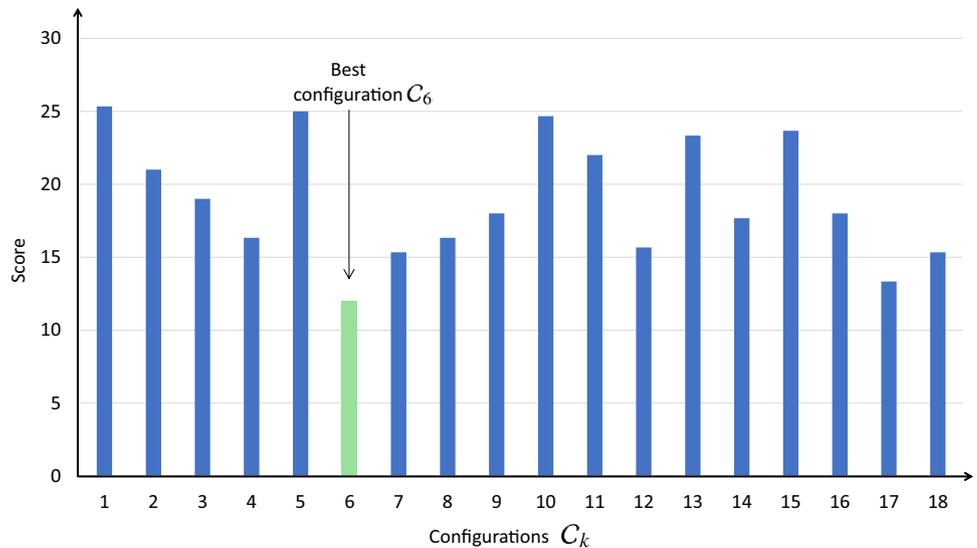
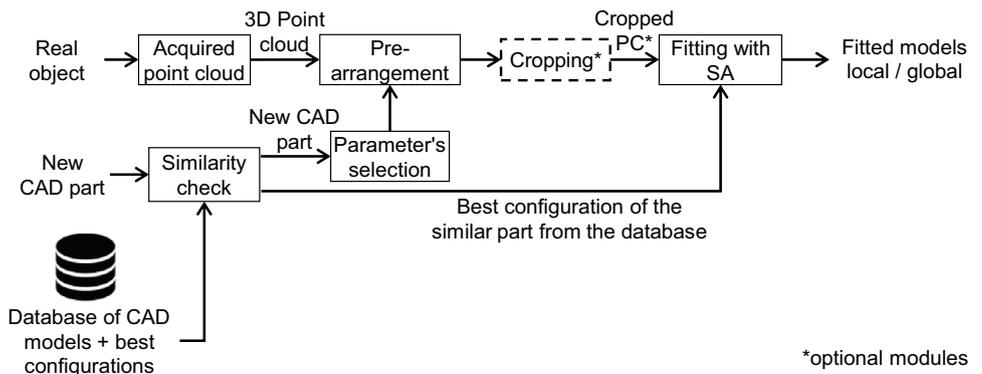


Fig. 9 Overall framework for fitting a new CAD model using the best configuration of a similar CAD model found in the database



\*optional modules

will tend to produce results depicting low deviations for the parameters  $p_k$ , low overall final energy as well as a limited number of iterations.

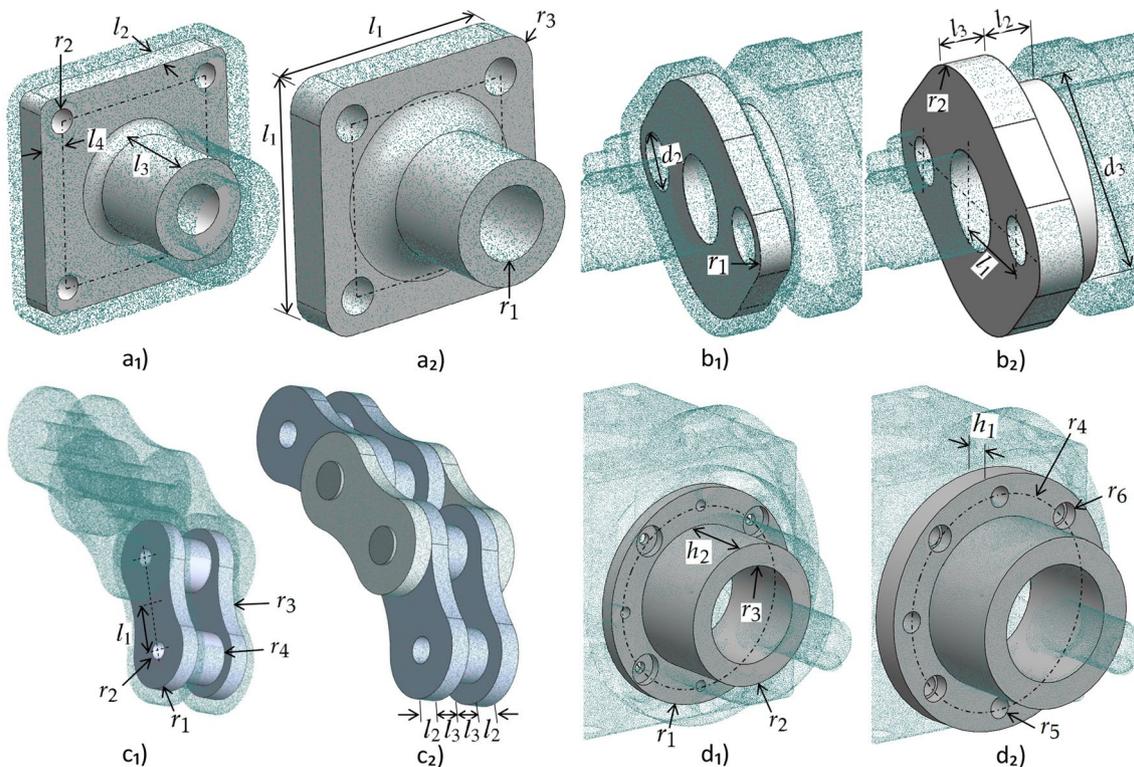
## 5 Results and discussion

The core of the fitting algorithm has been implemented in MATLAB®, which is able to call the built-in functions of SolidWorks® to perform the successive CAD model updates and ensure the consistency of the resulting B-Rep model during the optimization loops. It also handles the tasks of performing the sensitivity analysis of parameters and their grouping using K-mean clustering technique. For the ease of users, a prototype software created in VB script has been integrated as a plugin in SolidWorks® 2017 Education Edition [5] that allows efficient implementation of the proposed fitting technique.

### 5.1 Creation of a database of CAD models

To create the database and start populating it with CAD models, 17 CAD models have been selected (Fig. 14). Due

to space limitation, in this paper, only four test cases from the database are detailed, but the similarity assessment is performed on all of them. A rectangular flange, a gland, a chain link and a round flange, all of them being fitted several times to their respective point clouds using our SA-based fitting algorithm through DoEs (Sect. 4.6). The considered CAD templates before and after the fitting with their best configurations are shown in Fig. 10. The point clouds for the rectangular flange, chain assembly and the gland are virtually generated using as-scanned point cloud generation technique [22]. The proposed approach starts by generating a triangle mesh wrapping the CAD assembly model to be reverse-engineered. The resulting watertight mesh is then sampled to obtain a more realistic distribution of points. The occlusion phenomenon is then simulated using a hidden point removal algorithm launched from several viewpoints. A misalignment procedure can optionally be used to simulate the fact that in real-life acquisitions the position and orientation of the laser scanner and/or real object would have been changed to get a different scanning viewpoint. The virtual scanning process ends by generating noise and by inserting outliers. Global fitting is to be considered for the rectangular flange, and local fitting for the gland and



**Fig. 10** Four CAD models before and after the fitting with their best configuration resulting from the DoEs: (a<sub>1</sub>) coarse pre-arrangement of the rectangular flange in the virtually-generated point cloud; (a<sub>2</sub>) globally fitted rectangular flange; (b<sub>1</sub>) pre-arrangement of the gland; (b<sub>2</sub>) locally fitted gland in the virtually-generated point cloud; (c<sub>1</sub>)

pre-arrangement of the chain link; (c<sub>2</sub>) locally fitted chain link in the virtually-generated point cloud of a chain assembly; (d<sub>1</sub>) pre-arrangement of the round flange; (d<sub>2</sub>) locally fitted round flange in the real scanned point cloud of a gearbox assembly

chain link. The fourth example considers the local fitting of a round flange into a real point cloud obtained from scanning a gearbox assembly (Fig. 11a). Here, the point cloud has been acquired using a ROMER Absolute Arm 7520 SI, and while scanning the assembly as a whole, i.e. without disassembling it prior to its digitalization. The raw point cloud has been pre-processed before entering the reconstruction steps (i.e. noise and outliers removal, registration, filtering). By the end of this step, a clean point cloud having about 180k points is obtained (Fig. 11b).

Once the CAD models are pre-arranged (Fig. 10a<sub>1</sub>, b<sub>1</sub>, c<sub>1</sub>), their sensitivities are computed at prearranged position according to steps detailed in Sect. 4.3. All the steps  $h_k$  used to compute the sensitivities are equal and only depend on the type of parameter they are associated to. Thus, step  $h_k = 0.3$  mm in case  $p_k$  is a length, and  $h_k = 0.3^\circ$  in case it is an angle. Those values have been chosen in coherence with the accuracy of the adopted CAD modeler. For the sensitivities at prearranged position  $S_{k,0}$ , the forward finite difference scheme  $S_{k,j}^+$  has been used. After sensitivity analysis at the prearranged position, the control parameters  $p_k$  are rearranged in a decreasing order starting from the most sensitive to the least sensitive one (Fig. 5), and this is repeated for all the CAD models at the prearranged positions. Then, the parameters are grouped according to the different criteria detailed in Sect. 4.4. For all the CAD models used for the creation of the database, Table 5 shows the grouping of parameters according to either the level of sensitivities or the type of features they belong to.

The next step is to fit the CAD models with the SA-based fitting algorithm using DoEs to test the 18 configurations of setting parameters and identify the best configuration  $C_k$

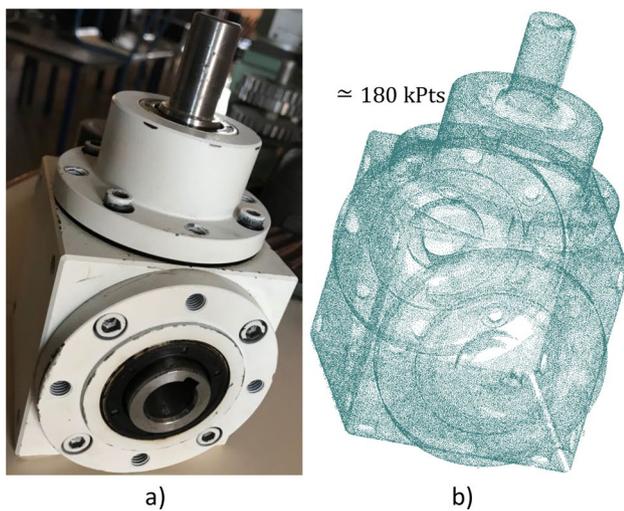


Fig. 11 Point cloud **b** obtained while scanning a real gearbox **a** without disassembling it and using a ROMER Absolute Arm 7520 SI laser scanner

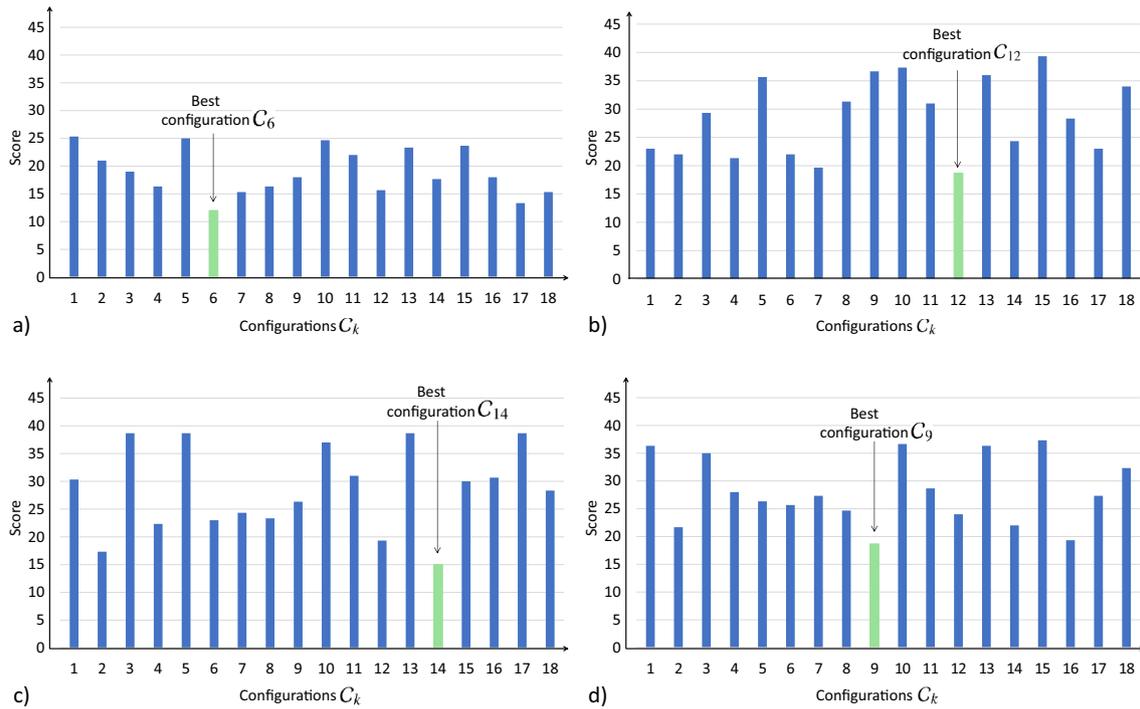
Table 5 Grouping of parameters  $p_k$  according to their sensitivity levels and to the types of feature they refer to

Groups	According to sensitivities using K-means	According to features
Rectangular flange		
$G_1$	$l_2, l_3, r_1$	$l_2, l_3, r_1, l_1$
$G_2$	$l_1$	$l_4, r_2$
$G_3$	$r_3, r_2, l_4, r_4$	$r_3, r_4$
Gland		
$G_1$	$l_3, l_2, r_2, r_1, l_1$	$l_3, l_2, r_2, r_1, d_3$
$G_2$	$d_3, d_2$	$l_1, d_2$
Chain link		
$G_1$	$r_1, r_4, l_2$	$r_1, r_4, l_2, l_1, l_3$
$G_2$	$l_1, l_3, r_2$	$r_2, r_3$
$G_3$	$r_3$	–
Round flange		
$G_1$	$h_1, r_1, h_2, r_2$	$h_1, r_1, h_2, r_2$
$G_2$	$r_4, r_5$	$r_4, r_5, r_6, d_1$
$G_3$	$r_6, d_1$	–

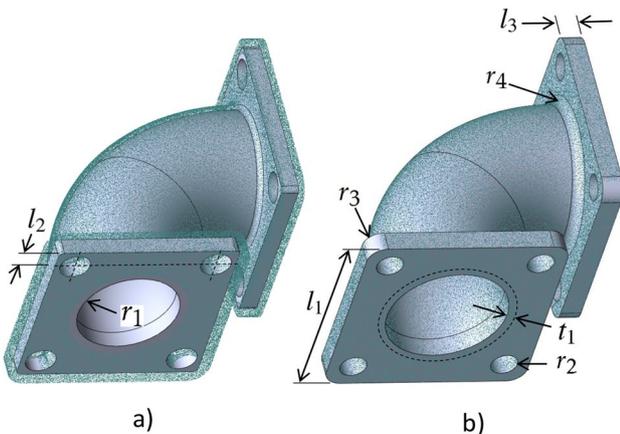
for each CAD model. The rectangular and round flanges have 8 control parameters that drive their shape during the optimization process, while the gland and the chain link are controlled by 7 parameters each, as visible in Fig. 10 that also shows the fitting results using best configurations. All the configurations  $C_k$  of parameters used for the fitting are listed in Table 4. After completing all the experiments, the best configuration of each CAD model is identified by selecting the configuration with the lowest sum of scores between the resulting final energy, the sum of absolute deviations of parameters  $p_k$  and the total number of iterations. Obtained results are depicted in Fig. 12 and clearly show the stochastic behavior of SA that is not very consistent for the fitting of all four models and cannot be correlated. To smooth this effect, each set of DoEs has been run three times and mean scores are taken. The average final obtained values, the averages absolute deviation and the average relative deviation between the point cloud and the CAD models are listed in Table 6. All the identified best configurations 6, 12, 14, and 9 are stored in the database together with the related CAD models. The corresponding setting parameters are listed in Table 4.

### 5.2 Fitting new CAD templates using stored configurations of similar parts

This section details how a new CAD template can be fitted using the stored best configuration of the most similar part retrieved by performing a similarity analysis. This is illustrated on the fitting of a bend joint (Fig. 13). The global similarity assessment has been performed between the



**Fig. 12** Scores obtained from averaging the results obtained from DoEs performed 3 times for the selection of best configuration of: **a** rectangular flange; **b** gland; **c** chain link and **d** round flange



**Fig. 13** Global fitting of bend joint defined by eight control parameters: **a** initial configuration after the prearrangement step, **b** final fitted part minimizing the deviation with the point cloud and using best configuration  $C_6$

bend joint (considered as query model) and the 17 models in the database using the tool of Giannini et al. [21]. They adopted an attributed graph containing both geometrical and topological information to represent a model. A correspondence among two models is found by the identification of the maximum common subgraph between the two attributed graphs representing the models. This problem is solved transposing to the maximum clique (MC) problem

employing an association graph in which the arcs linking the associated faces vary according to the search objectives. To detect the MC, a simulated annealing-based minimization is used to minimize the function characterizing the clique property. Figure 14 shows all the CAD models in the database used for similarity check and the results of this analysis on the new bend joint is presented in Table 7. The number of corresponding nodes and similarity measure for each model are also listed. The query model is composed of 34 faces, of which 10 are planar, 10 cylindrical, 10 fillets, and the 4 are sweep parallel faces of the bend. Among the models in database, the rectangular flange presents the same pattern and is returned with a similarity measure value of 53% (max = 100). The other models present only a portion of the specified pattern and are correctly returned with smaller similarity measure values.

Having the highest similarity measure for the bend joint, the best configuration of the rectangular flange, i.e.  $C_6$  is used for the fitting of the bend joint. The bend joint whose shape is controlled by 8 parameters is prearranged into the virtually-generated point cloud to initialize the fitting process (Fig. 13a). Here, the control parameters  $p_k$  for the bend are used in a single group without any weighting strategy. The rest of the setting parameters of configuration  $C_6$  are highlighted in green in Table 4. Figure 13b shows the fitted bend joint part that has been reconstructed using the best configuration  $C_6$  of the rectangular flange.

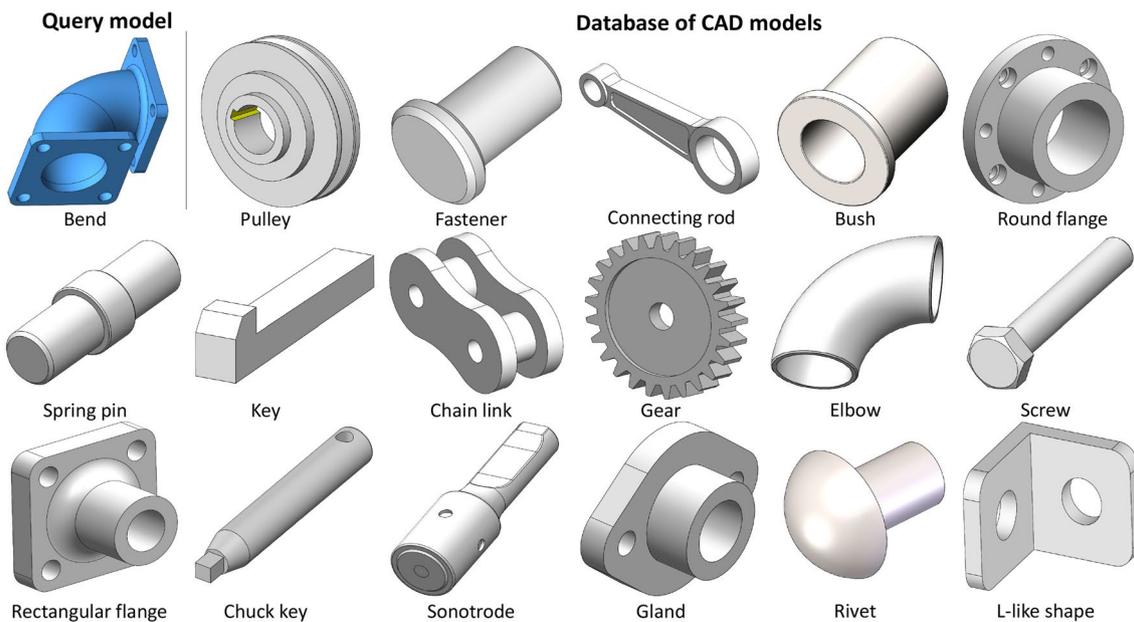
**Table 6** The average absolute and relative errors measured against the best configurations of four CAD models

Groups	$p_k$	$p_k^0$ (mm)	$p_k^D$ (mm)	$\overline{p_k^F}$ (mm)	$\overline{\delta p_k^F}$ (mm)	$\overline{\Delta p_k^F}$
Rectangular flange						
$G_1$	$l_2$	5	7	6.8625	0.1375	0.0196
	$l_3$	15	20	20.7677	0.7843	0.0392
	$r_1$	4	6	6.0957	0.1963	0.0327
	$l_1$	35	40	39.7657	0.2343	0.0059
	$r_3$	2	5	3.6618	1.3382	0.2676
	$r_2$	2	3	3.0765	0.0994	0.0331
	$l_4$	2	3	2.8167	0.1976	0.0659
	$r_4$	2	5	4.1763	0.8237	0.1647
Gland						
$G_1$	$l_3$	9	8	8.2922	0.2922	0.0365
	$l_2$	9	9	8.5431	0.4569	0.0508
	$r_2$	26	25	24.9621	0.0428	0.0017
	$r_1$	12	10	10.7650	0.7650	0.0765
	$d_3$	33	32	37.4636	5.4636	0.1707
$G_2$	$l_1$	23	24	23.2546	0.7454	0.0311
	$d_2$	9	7	6.6266	0.3734	0.0533
Chain link						
$G_1$	$r_1$	62	63	62.6977	0.5043	0.0080
	$r_4$	35	36	35.8767	0.1233	0.0034
	$l_2$	28	27	27.0850	0.2617	0.0097
	$l_1$	68	72	72.3493	0.3493	0.0049
	$l_3$	27	27	27.0443	0.2677	0.0099
	$r_2$	16	18	18.3500	0.3500	0.0194
	$r_3$	170	162	174.1033	12.1033	0.0747
Round flange						
$G_1$ (sens)	$h_1$	7	10.1	9.8534	0.2466	0.0244
	$r_1$	42	44.5	44.8796	0.3796	0.0085
	$h_2$	30	30	30.1667	0.1667	0.0056
	$r_2$	28	30	29.9211	0.0789	0.0026
$G_2$ (sens)	$r_4$	36	37	37.3740	0.3740	0.0101
	$r_5$	3	4	3.7290	0.2710	0.0678
$G_3$ (sens)	$r_6$	5	5	5.5578	0.5578	0.1116
	$d_1$	3	4	5.8473	1.8473	0.4618

To evaluate and validate the assumption that best configurations can be used for similar parts, the bend joint is then fitted using the DoEs strategy to identify its best configuration as previously done for the other CAD templates of the database. The same 18 configurations were tested for the bend joint and repeated three times to take the average of the final obtained scores. The resulting scores for all the configurations are shown in Fig. 15. The best configuration for the bend joint is also the same as for the rectangular flange, i.e. configuration  $C_6$ . This validates the idea that new models to be reconstructed can make use of the best configuration of the similar CAD models to produce good quality fitting.

## 6 Conclusion

This paper has explored the possibility to tune the setting parameters of the SA-based fitting algorithm used for fitting parametrized CAD models locally and globally in the point cloud of a digitized part or an assembly. A framework integrating sensitivity analysis of the parameters has been proposed to understand how the shape driving parameters of the CAD models are modified by the SA algorithm and how such an approach helps to group the CAD model's parameters to treat batches having comparable sensitivities. Using the DoEs approach, the framework also details the steps for the identification of the best configuration of



**Fig. 14** Similarity check between the query model and the models in database

**Table 7** Similarity check for the bend part with models in the database

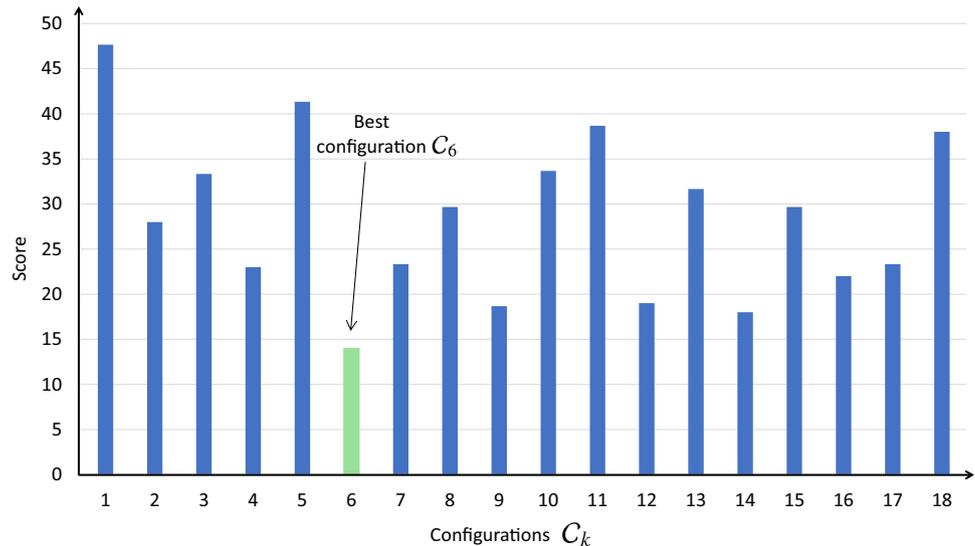
CAD Models database	Nodes	Similarity (%)
Pulley	07	14
Fastener	02	03
Connecting rod	15	23
Bush	03	04
Spring pin	02	03
Key	07	27
Chain link	08	18
Gear	22	08
Elbow	02	04
Round flange	15	49
Screw	07	20
Rectangular flange	15	53
Chuck key	06	14
Sonotrode	22	46
Gland	14	52
Rivet	02	04
L-like shape	11	35

setting parameters for the SA algorithm that will enhance the quality of the fitting approach. A framework including the generation of a database of CAD models stored with their best configurations has also been presented. Such a

database of CAD models will help the users to make use of those stored best configurations for the new parts to be fitted by performing a similarity check between the new models and the models in the database.

At the moment, chosen parameters are user-specified and are linked to functional meaning, but a preprocessing step could be introduced in the optimization loop to decrease the interdependency between the parameters. For thin parts like sheet metal where the thickness is too small, the number of as-scanned points on such features will also be fewer. Consequently, such parameters will not contribute that much to the overall energy and may be neglected by SA for better fitting. For such thin parts, working on the 2D sketches would be more appropriate. Or, the considered parameters could be weighted knowing that it is a sheet metal part. The created database is rather small and only contains few 3D CAD parts. However, the overall framework has been validated. For future perspectives, the database can be enlarged to also contain parametric models and their best configurations for the fitting of 2D sketches and assemblies. Sensitivity analysis performed at the start of the optimization can be integrated into the SA algorithm to track the sensitivities of parameters all along the optimization. Keeping the track of the evolution of sensitivities, the system will automatically handle the less sensitive parameters.

**Fig. 15** Resulting scores for the 18 configurations used for the SA-based fitting of bend joint



## References

- Lu Y (2017) Industry 4.0: a survey on technologies, applications and open research issues. *J Ind Inf Integr* 6:1–10
- Falcidieno B, Giannini F, Léon J-C, Pernot J-P (2014) Processing free form objects within a product development process framework. *Adv Comput Inf Eng Res* 317–344
- Louhichi B, Abenham GN, Tahan AS (2015) CAD/CAE integration: updating the CAD model after a fem analysis. *Int J Adv Manuf Technol* 76(1):391–400
- Shah GA, Polette A, Pernot J-P, Giannini F, Monti M (2020) Simulated annealing-based fitting of CAD models to point clouds of mechanical parts' assemblies. *Eng Comput* 1–19
- Shah GA, Polette A, Pernot J-P, Giannini F, Monti M (2022) User-driven computer-assisted reverse engineering of editable CAD assembly models. *J Comput Inf Sci Eng* 22(2)
- Buonamici F, Carfagni M, Furferi R, Governi L, Lapini A, Volpe Y (2018) Reverse engineering of mechanical parts: a template-based approach. *J Comput Des Eng* 5(2):145–159
- Buonamici F, Carfagni M, Furferi R, Volpe Y, Governi L (2021) Reverse engineering by CAD template fitting: study of a fast and robust template-fitting strategy. *Eng Comput* 37(4):2803–2821
- Kirkpatrick S, Gelatt C, Vecchi M (1982) Optimization by simulated annealing. IBM Research Report RC 9355, Acts of PTRC Summer Annual Meeting
- Hutter F, Hoos HH, Leyton-Brown K, Murphy K (2010) Time-bounded sequential parameter optimization. In: International conference on learning and intelligent optimization. Springer, pp 281–298
- Gunawan A, Lau HC et al (2011) Fine-tuning algorithm parameters using the design of experiments approach. In: International conference on learning and intelligent optimization. Springer, pp 278–292
- Iooss B, Lemaître P (2015) A review on global sensitivity analysis methods. In: Uncertainty management in simulation-optimization of complex systems. Springer, pp 101–122
- Hamby DM (1994) A review of techniques for parameter sensitivity analysis of environmental models. *Environ Monit Assess* 32(2):135–154
- Jin Y, Meng X, Ziyou G (2009) Sensitivity analysis of simulated annealing for continuous network design problems. *J Transport Syst Eng Inf Technol* 9(3):64–70
- Gamboa F, Janon A, Klein T, Lagnoux A et al (2013) Sensitivity analysis for multidimensional and functional outputs. *Electron J Stat* 8(1):575–603
- Spagnol A, Riche RL, Veiga SD (2019) Global sensitivity analysis for optimization with variable selection. *SIAM/ASA J Uncertain Quantif* 7(2):417–443
- Lamboni M, Monod H, Makowski D (2011) Multivariate sensitivity analysis to measure global contribution of input factors in dynamic models. *Reliab Eng Syst Saf* 96(4):450–459
- Robinson TT, Armstrong CG, Chua HS, Othmer C, Grahns T (2012) Optimizing parameterized CAD geometries using sensitivities based on adjoint functions. *Comput Aided Des Appl* 9(3):253–268
- Zhan S-H, Lin J, Zhang Z-j, Zhong, Y-W (2016) List-based simulated annealing algorithm for traveling salesman problem. *Comput Intell Neurosci*
- Bellio R, Ceschia S, Di Gaspero L, Schaerf A, Urli T (2016) Feature-based tuning of simulated annealing applied to the curriculum-based course timetabling problem. *Comput Oper Res* 65:83–92
- Atiqullah MM, Rao S (2001) Tuned annealing for optimization. In: International conference on computational science. Springer, pp 669–679
- Giannini F, Lupinetti K, Monti M (2017) Identification of similar and complementary subparts in B-rep mechanical models. *J Comput Inf Sci Eng* 17(4)
- Montlahuc J, Shah GA, Polette A, Pernot J-P (2018) As-scanned point clouds generation for virtual reverse engineering of CAD assembly models. *Comput Aided Des Appl* 1–11
- (2021) Design of experiments via Taguchi methods-orthogonal arrays. University of Michigan. Online; Accessed 07 2022 Mar
- Roy RK (2010) A primer on the Taguchi method. Society of Manufacturing Engineers
- Gonzales GV, Dos Santos ED, Emmendorfer LR, Isoldi LA, Rocha LAO, Estrada E (2015) A comparative study of simulated annealing with different cooling schedules for geometric optimization of a heat transfer problem according to constructual design. *Sci Plena* 11(8):11
- Ingber L (1996) Adaptive simulated annealing (asa): lessons learned. *Control and Cybern* 25(1):32–54 (cited By 364)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.