Information Processing Letters 111 (2011) 857–863

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

HC_AB: A new heuristic clustering algorithm based on Approximate Backbone

Yu Zong [a,b,c], Guandong Xu [c], Ping Jin [a,*,1,2,3,4], Yanchun Zhang [c], Enhong Chen [b]

[a] Department of Information and Engineering, West Anhui University, Luan, 237012, China
[b] Department of Computer Science and Technology, University of Science and Technology of China, Hefei, 230026, China
[c] Centre for Applied Informatics, Victoria University, Melbourne, VIC8001, Australia

ARTICLE INFO

Article history:
Received 15 September 2009
Received in revised form 28 May 2011
Accepted 30 May 2011
Available online 15 June 2011
Communicated by F.Y.L. Chin

Keywords:
Approximation algorithms
Data mining
Heuristic clustering
NP-hard problem
Approximate Backbone

ABSTRACT

Clustering is an important research area with numerous applications in pattern recognition, machine learning, and data mining. Since the clustering problem on numeric data sets can be formulated as a typical combinatorial optimization problem, many researches have addressed the design of heuristic algorithms for finding sub-optimal solutions in a reasonable period of time. However, most of the heuristic clustering algorithms suffer from the problem of being sensitive to the initialization and do not guarantee the high quality results. Recently, *Approximate Backbone* (AB), i.e., the commonly shared intersection of several sub-optimal solutions, has been proposed to address the sensitivity problem of initialization. In this paper, we aim to introduce the AB into heuristic clustering to overcome the initialization sensitivity of conventional heuristic clustering algorithms. The main advantage of the proposed method is the capability of restricting the initial search space around the optimal result by defining the AB, and in turn, reducing the impact of initialization on clustering, eventually improving the performance of heuristic clustering. Experiments on synthetic and real world data sets are performed to validate the effectiveness of the proposed approach in comparison to three conventional heuristic clustering algorithms and three other algorithms with improvement on initialization.

1. Introduction

Clustering is a process of grouping a set of objects into clusters so that objects within a cluster are highly similar, but are dissimilar to objects in other clusters [1–3]. A common way of clustering is to calculate a set of centres such that the sum of squared errors between data objects and their nearest centres are minimized [4]. Essentially, this is equivalent to a classical combinatorial optimization problem, but solving it completely is NP-hard even with a simplest case of just two clusters [5]. The solution becomes worse in case of a larger data set due the inherence of optimization problem. To address this, researchers are seeking heuristic methods to solve this problem. For example, K-centre clustering is a popularly used clustering approach based on heuristic search, which is to minimize the accumulated square errors in a designated region (i.e., guarding the heuristic search with the specified initial values). In this scenario, the obtained clustering result is dependent on the initialization and the minimized squared error only reflects the sub-optimal solution with this running of heuristic search. In the other words, the nature of heuristic search process makes the K-centre clustering algorithms heavily sensitive to the initialization settings, thus not guaranteeing the higher quality clustering results with randomly chosen initializations [4]. Therefore, how to deal with the sensitivity problem of initialization in K-centre clustering is becoming an active and well concerned

* Corresponding author. Tel.: +86 564 3305582.
E-mail addresses: nick.zongy@gmail.com, jinping@wxc.edu.cn (P. Jin).

[1] Supported by the key program of NSF 60933013.
[2] Supported NSF grant 60775037.
[3] Supported by ARC grant DP0770479.
[4] Supported by NSF of Anhui Education Department grant KJ2009A54 and KJ2011Z321.

0020-0190/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.ipl.2011.05.020

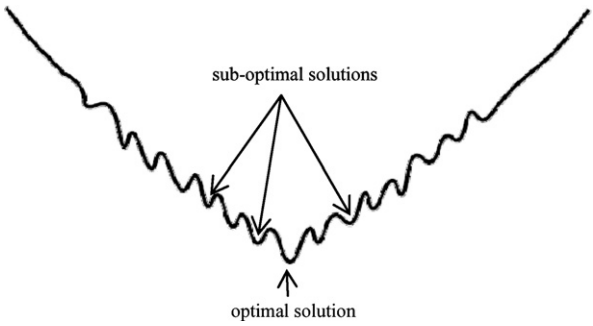

Fig. 1. Illustrative picture of the "big valley" phenomenon.

challenge in clustering research community. As various executions of K-centre clustering with different initializations capture the different sub-optimal solutions corresponding to different searching regions, it is intuitive that combining these sub-optimal solutions into consideration will greatly facilitate the identification of the optimal information in the whole search range. This intuition originates a variety of new clustering algorithms such as cluster ensemble [6], consensus clustering [7] and our approach proposed in this paper.

On the other hand, we from the various executions of K-centre algorithm can observe that the different sub-optimal clustering results do reflect the existences of data objects gathering around the different cluster centres in a data set [6]. If we can make full use of these sub-optimal results for the determination of the initialization, a more appropriate start in heuristic search will be obtained, making that the sensitivity problem of initialization effectively tackled. Furthermore, in [8–10], the authors have indicated an interesting phenomenon that nearly 80% of the sub-optimal results are distributed around the optimal result when they dealt with the Traveling Salesman Problem (TSP) by using heuristic search. Inspired by their findings (i.e. the "big valley" phenomenon shown in Fig. 1), we further envision the use of the overlapped common intersections of these sub-optimal results as a priori, to help the convergence to the optimal solution during the optimization process. Particularly in the context of K-centre clustering, it is expected that properly choosing these intersection areas as the initialization would benefit overcoming the sensitivity of initialization and finding the better clustering result.

Backbone analysis is becoming an active research topic in NP-hard problems recently. The Backbone defined as the core part of all optimal solutions was first proposed in [11] for TSP and it has been successfully applied in different applications [12,13]. A complete Backbone, however, is usually impossible to be obtained for many optimization problems in real applications. Instead, *Approximate Backbone* (AB), an approximate form of Backbone, i.e. the intersection of different sub-optimal solutions of a dataset, is becoming a practical means in real applications. The AB is often used to investigate the characteristics of a dataset and expedite the convergence speed of heuristic algorithms [14–16].

Inspired by the above discussion on the use of sub-optimal results and the concept of AB, we in this paper intend to introduce the AB to address the initialization problems in heuristic clustering algorithms described above, and in particular, we propose an algorithm named *Heuristic Clustering Approach Based on Approximate Backbone* (HC_AB). The basic process of HC_ AB is that: we first identify the AB from a set of sub-optimal solutions derived by running K-centre clustering with different initialization settings; then, we construct a new initialization based on the AB for heuristic search; eventually, we re-run the K-centre clustering algorithm by using this new initialization to generate a better clustering result. Experiments on synthetic and real world data sets have been conducted to validate the effectiveness of the proposed approach on improving the quality of clustering and reducing the impact of initialization.

As mentioned above, HC_AB essentially follows the similar fundamental principle of cluster ensemble and consensus clustering [6,7] that makes use of the multiple clustering results, but with different focuses. Specially, Approximate Backbone captures the intersection of data objects within the clusters derived from multiple execution of clustering, from the perspective of the original significance of data to form a new initialization, and then re-run the K-centre clustering algorithm again on the data set with this new initialization to generate the better clustering result. In contrast, cluster ensemble and consensus clustering directly assemble the various clustering results in a unified manner to get the final cluster result rather than re-running the K-centre clustering algorithm.

In summary, the main contributions of this paper are (1) we define the intersection of various sub-optimal solutions, i.e. the AB, as the new search start point for heuristic clustering; (2) we define the quality measures, namely scale and purity to guide the selection of AB and propose a new algorithm to address the heuristic clustering; (3) we conduct experiments to evaluate the efficiency and effectiveness of the proposed approach.

The rest of paper is structured as follows: Section 2 gives the theoretical background of this paper, Section 3 discusses the algorithmic details and Section 4 reports the experimental results and comparisons. Section 5 concludes the paper.

2. Theoretical background of Approximate Backbone

In this section, we first briefly discuss the concept of optimal clustering result and the sub-optimal clustering result, and then, we give the definition of Backbone and Approximate Backbone.

Given a data set $D = \{x_1, x_2, \ldots, x_N\}$ which contains N objects and each object $x_i \in D$ is described by d numeric attributes. Let $dist : R^d \times R^d \mapsto R_+$ be a given distance function between any two objects in R^d. The clustering problem on numeric data sets can be formulated as: partitioning the N objects into distinct K clusters such that the overall distance function $\phi = \sum_{k=1}^{K} \sum_{x_i \in C_k} dist(x_i, v_k)$ is minimized, where C_k is a cluster and v_k is the centre of C_k. For simplicity, here we denote each cluster centre v_k by one object identifier in D, for example, 22, 78 represent the cluster centre being the data point #22, #78 and so on. As discussed above, this clustering problem is

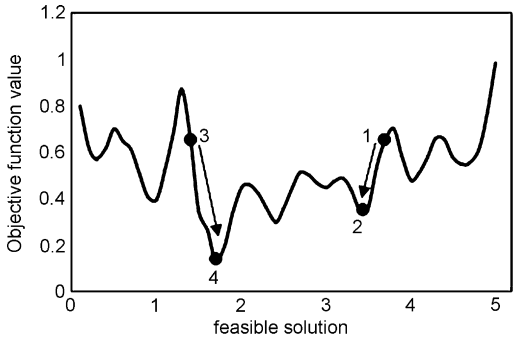

Fig. 2. An example of K-centre clustering algorithm with different initializations.

actually a typical combinatorial optimization problem. The search space S of this clustering problem consists of all the possible combinations of the data objects. For optimization, we need to traverse S to find out a set of data objects $V^* = \{v_1^*, \ldots, v_K^*\}$ such that the ϕ value is minimized. This data objects set $V^* = \{v_1^*, \ldots, v_K^*\}$ is defined as the optimal clustering result. Obviously, it is almost impractical to thoroughly traverse the S of a very large data set due to the NP-hard nature. Recently, to deal with time consumption many researchers have proposed heuristic clustering algorithms that only search a subset $S' \subset S$ to discover the approximation of the optimal solution. And the clustering result $V = \{v_1, v_2, \ldots, v_K\}$ corresponding to the smallest ϕ value in S' is considered as the sub-optimal clustering result. For example, K-centre clustering algorithm is a typical heuristic clustering algorithm, which takes D as an input to achieve a sub-optimal clustering result with an initialization. Fig. 2 gives an illustration of how the different sub-optimal solutions are achieved with different initializations in K-centre clustering algorithm. In Fig. 2, the X axis denotes feasible clustering results and the Y axis denotes the corresponding ϕ values of feasible solutions. Without loss of generality, given that the point 3 is selected as an initialization for K-centre clustering algorithm, it will converge to the point 4, which is one of the feasible solutions with optimal ϕ value. If the optimization process starts from the point 1 for the same clustering algorithm, the algorithm will reach to another optimal result of point 2, which is actually a worse solution with larger ϕ value.

The multiple executions of K-centre clustering with various initialization settings form a collection of sub-optimal clustering results, and our aim is to utilize the intersection of these sub-optimal solutions to construct a new good core for a search start point. Since the new search start point does reflect the closeness to the optimal solution, the heuristic clustering algorithm is much able to obtain a clustering result close to the optimal one. In this manner, the initialization sensitivity problem is considerably handled and the heuristic clustering quality is accordingly improved. To further consolidate the theoretical foundation of our intuition, we below define the following concepts.

Definition 1 *(Backbone).* Given a collection of optimal clustering results for a numeric clustering problem i.e. $Z^* = \{V^{1*}, V^{2*}, \ldots, V^{P*}\}$, in which each optimal result V^{p*} is denoted by $V^{p*} = \{v_1^{p*}, v_2^{p*}, \ldots, v_K^{p*}\}$, $p = 1, \ldots, P$, the *Backbone* of this clustering problem is defined as the intersection of all optimal clustering results:

$$backbone\big(V^{1*}, V^{2*}, \ldots, V^{P*}\big) = V^{1*} \cap V^{2*} \cap \cdots \cap V^{P*}.$$

Principally, the optimal solution is hard to obtain for an NP-hard problem in reality, resulting in the difficulty in identifying the theoretically ideal Backbone. However, in some studies, researchers have observed an interesting phenomenon that there are nearly 80% sub-optimal solutions being distributed around the optimal solution and a "big valley" structure is seen [8–10]. Motivated by this observation, we intuitively have an idea in mind on how to approximate the ideal Backbone by making use of the sub-optimal solutions [12].

Definition 2 *(Approximate Backbone).* Given a collection of sub-optimal clustering result of a numeric clustering problem, i.e. $Z = \{V^1, V^2, \ldots, V^M\}$, in which each sub-optimal result V^m is denoted by $V^m = \{v_1^m, v_2^m, \ldots, v_K^m\}$ $m = 1, \ldots, M$, the *Approximate Backbone* (AB) is defined as the intersection of all sub-optimal clustering results:

$$a\_bone\big(V^1, V^2, \ldots, V^M\big) = V^1 \cap V^2 \cap \cdots \cap V^M.$$

As described above, our method aims to make use of the AB of sub-optimal solutions to form the initialization (i.e. the start point for heuristic search), thus constructing an appropriate AB in K-centre clustering for heuristic search becomes an important issue. In other words, the quality of the K-centre clustering results is greatly dependent on the characteristics of AB. Essentially, the size and quality of AB are two key measures needed to be considered, here we propose two parameters to describe the characteristics of AB—*Scale* and *Purity*. The former one describes how many percentages of total sub-optimal solutions are included in the AB; whereas the latter one denotes how many percentages of sub-optimal solutions included in the AB also exist in the theoretically ideal Backbone simultaneously. In particular, *Approximate Backbone Scale* (*ABS*) and *Approximate Backbone Purity* (*ABP*) are defined as follows.

Definition 3 *(Approximate Backbone Scale).* Given an AB, $a\_bone(V^1, V^2, \ldots, V^M)$, the *Approximate Backbone Scale* is defined as the ratio of the AB's cardinality to the cluster number K:

$$\mathrm{ABS} = \frac{|a\_bone(V^1, V^2, \ldots, V^M)|}{K}.$$

Definition 4 *(Approximate Backbone Purity).* Given an AB, $a\_bone(V^1, \ldots, V^M)$, and a backbone $backbone(V^{1*}, \ldots, V^{P*})$, the *Approximate Backbone Purity* is defined as the ratio of the cardinality of the intersection of the AB and the Backbone to the AB's cardinality:

$$\mathrm{ABP} = \frac{|a\_bone(V^1, V^2, \ldots, V^M) \cap backbone(V^{1*}, V^{2*}, \ldots, V^{P*})|}{|a\_bone(V^1, V^2, \ldots, V^M)|}.$$

Table 1
An example of clustering results of D.

Name	Centre set
V^*	22, 78, 109, 180, 230, 292, 310, 366, 412, 475
V^1	43, 78, 109, 198, 240, 262, 310, 366, 412, 480
V^2	43, 78, 128, 198, 240, 262, 310, 366, 412, 480
V^3	43, 78, 128, 198, 240, 252, 310, 366, 432, 480

In this paper, we assume that the originating cluster centres indicate the unique optimal solution that should be found by clustering, thus we here particularly treat these originating cluster centres as the "ideal" Backbone to calculate the purity of Approximate Backbone.

In order to achieve the better result of heuristic clustering algorithm, we expect to form an appropriate AB with both large ABS and ABP values. The rationale behind this consideration is due to the two-fold assumptions. On the one hand, the larger the ABS values, the more proportions of the sub-optimal solutions are included in the AB, i.e. the higher optimal coverage. On the other hand, the bigger the ABP value, the better the included optimal solutions are, i.e. the included sub-optimal solutions are closely scattered around the optimal clustering result.

Thus determining the appropriate values of ABS and ABP becomes a crucial task in our proposed approach. However, in real applications, ABS and ABP possess totally different characteristics, resulting in the difficulty in empirically determining them. Below let's take an example to demonstrate the underlying relationship between ABS and ABP.

Consider a data set D containing 500 objects which form 10 clusters, each cluster is represented by a representative data object (i.e. cluster centre). The assumed optimal clustering result V^* (e.g. the real cluster centers) is shown at the first row in Table 1, and three sub-optimal clustering results V^1, V^2, V^3, obtained by running the K-centre clustering algorithm with three different initializations, are also listed in Table 1. There is only one optimal clustering result in D, that is $backbone(V^*) = V^*$.

For this example, we can obtain the AB: $a\_bone(V^1, V^2, V^3) = \{43, 78, 198, 240, 310, 366, 480\}$. Known from the definition of ABS, the value of ABS in this example is calculated as:

$$\text{ABS} = \frac{|a\_bone(V^1, V^2, V^3)|}{K} = \frac{7}{10} = 0.7.$$

We observe that there are three commonly overlapped centres existing in the AB and the Backbone, thus the value of ABP is:

$$\text{ABP} = \frac{|a\_bone(V^1, V^2, V_3^3) \cap V^*|}{|a\_bone(V^1, V^2, V^3)|} = \frac{3}{7} = 0.429.$$

According to Definition 2, the AB is derived from M sub-optimal solutions, so the characteristics of AB has a close correlation to M. In order to illustrate this relationship, we construct three data sets: RandomS1, RandomS2 and RandomS3, each of which contains 34 clusters. Each cluster has 100 data objects, among which 99 objects are generated by a Gaussian distribution function with different mean (μ) and standard deviation (σ) and the 100th data one is the mean of the rest of 99 data objects, i.e. the 100th data objects is the centre of the cluster in this manner. We run the Vertex Substitution Heuristic (VSH) algorithm [17], a classical K-centre clustering algorithm, on these three data sets, and denote the results as VSH_RandomS1, VSH_RandomS2 and VSH_RandomS3, respectively. VSH is executed for $M = 2 : 2 : 20$ times on each data set, where $M = 2 : 2 : 20$ means M changing from 2 to 20 with step 2. The relationships between ABS, ABP and M are shown in Fig. 3.

From Fig. 3, we can see that the changes on ABS and ABP are in opposite trends with M. ABS is gradually declining with the increase of M, while on the contrary, ABP is increasing with M, and eventually the changes of ABS and ABP become slight until a stable state is reached. The explanation to this observation is intuitively because with the increase of M, the overlap size of optimal results (ABS) would decrease due to the stricter joint operation requirement, while for ABP the likelihood of more optimal results

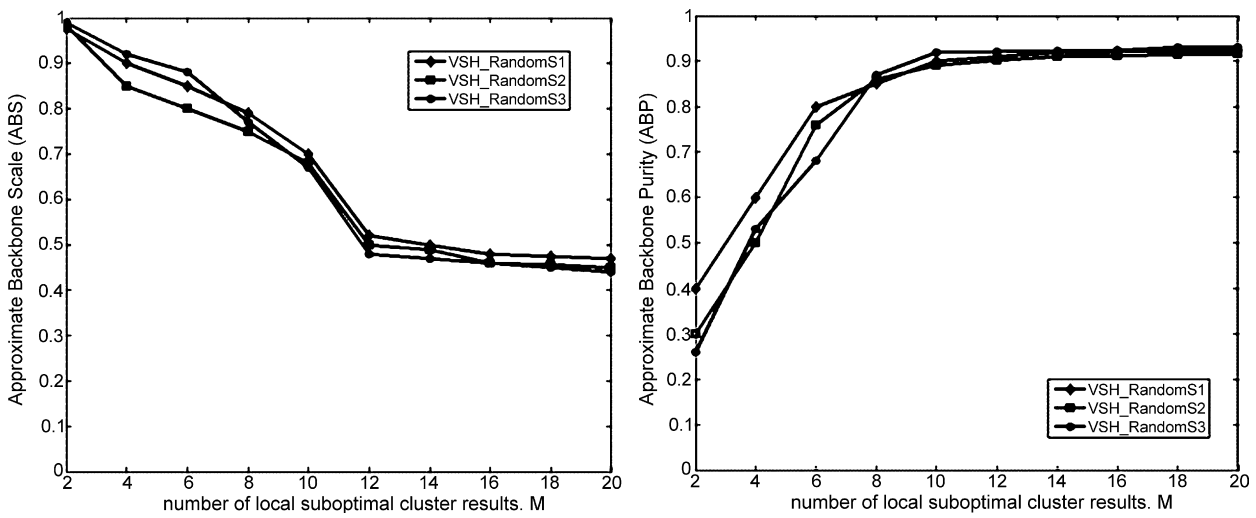

Fig. 3. The plots of ABS and ABP of AB.

in existence of the optimal clustering results dominates the change. As a consequence, ABS has a totally reverse change trend with M against ABP. As indicated above, we aim to form an appropriate AB with both higher ABS and ABP values to construct a good start point in heuristic clustering. Due to the reverse changes of ABS and ABP with M, we obviously are not able to achieve their best optimizations at the same time, and thus having to choose a tradeoff between them to ensure the better initialization. In real applications, the tradeoff of ABS and ABP is determined by selecting an appropriate M. However, the setting of M is still an open question in various applications [14,15], the well used approach of selecting the M value is through empirically choosing a reasonable M, which can guarantee a better AB found. In the experimental study part, we will describe this process of selecting M in detail.

3. Heuristic clustering algorithm based on Approximate Backbone

In this section, we present the description of our proposed algorithm as shown in Algorithm 1 which gives the pseudo code of HC_AB. The whole process consists of two stages, namely (1) the new initialization construction and (2) the re-execution of K-centre clustering.

Algorithm 1: HC_AB.

Input: D, K, M
Output: *best_V*
(1) Generate M clustering result, $Z = \{V^1, V^2, \ldots, V^M\}$, by running K-centre clustering algorithm;
(2) Find the AB $a\_bone(V^1, V^2, \ldots, V^M)$ from Z;
(3) Select $K - |AB|$ centres from ST;
(4) Rerun K-centre clustering algorithm with the new initialization;
(5) Return *best_V*.

In first stage, a K-centre clustering algorithm is run on D with different initializations to generate the sub-optimal clustering results collection Z. The AB $a\_bone(Z)$ is generated based on Definition 2 (shown in step 2) in order to construct the first part of initialization centres. Since $a\_bone(Z)$ only has $|AB|$ centres, which is less than the predefined cluster number K, we need to find the rest $K - |AB|$ initial centres. As shown in step 3, we then do the search in $ST = \bigcup Z \setminus a\_bone(Z)$ to determine the $K - |AB|$ points with the highest frequency of presence in the estimated optimal results Z, as the complementary part of the initialization centres. The reason behind this computation is how to select the $K - |AB|$ points from the $C_{|ST|}^{K-|AB|}$ possible results in ST via an appropriate way, which is usually a time-consuming process. To deal with it, here we adopt a greedy method. Because the K-centre clustering algorithm could generate well dispersed clustering results, we intend to select the $K - |AB|$ centres with the highest presence frequencies in Z and the biggest deviations from the AB. To do this, for each centre in ST, we first calculate the average distance between the selected centre and all other centres in the AB, and then multiply the average distance with its presence frequency to generate a value for this centre. The values of all candidate centres in ST form a sequence; we sort it and select the top $K - |AB|$ centres. Finally we combine these $K - |AB|$ centres with the $|AB|$ centres derived from AB to generate a new initialization setting and re-execute the K-centre clustering algorithm with this new initialization setting to generate the final clustering result *best_*V (step 4).

According to the above discussion, each centre in Z is uniquely represented by a data object ID in D, however, for some kinds of K-centre clustering algorithms, such as K-means where the clusters centres are the mean of all data objects in the cluster rather than one data object. In this case, we will use the follow steps to obtain the AB: (1) in the same way we run the K-means algorithm for M times on the data set D; and for each corresponding cluster of M times results we use the join operation on the data objects of the cluster to filter out a set of data objects that are co-occurred in the same cluster, i.e., for example, for the first cluster, we jointly filter out the data objects that simultaneously assigned to the first cluster in total M running; (2) we calculate the mean of these filtered data objects within the mth cluster as V_{km}^k, $m = 1, \ldots, K$, and eventually form the $AB = \{V_{km}^1, V_{km}^2, \ldots, V_{km}^K\}$, note that V_{km}^m could be void due to the no existence of data objects jointly occurred in M results, i.e., $K \geqslant |AB|$; (3) we treat this AB as one part of initialization and follow the same process to add the $K - |AB|$ objects to obtain the final initialization. For more details regarding this process, please refer to [18].

Table 2
Descriptions of datasets used in experiments.

Data set	Size	Attribute	Class
RandomS	2700	2	34
Statlog(Heart)	178	13	7
Digits	5620	64	9
Water	527	64	13
Sponge	76	45	12

4. Experiments

In order to evaluate the effectiveness of AB on overcoming the initialization sensitivity of K-centre clustering algorithms, we compare our algorithm with three traditional K-centre clustering algorithms: K-means [19], CLARANS [20] and VSH [17] and three modified K-centre clustering algorithms with the improvement on initialization: CCIA [21], CSI [22] and kd-tree [23], in terms of cluster quality. We assess the clustering results derived by these clustering algorithms on five data sets (shown in Table 2), including a random synthetic data set and four real world data sets downloaded from UCI. Particularly, in data pre-processing, we omit all the non-numeric attributes from Sponge, Statlog and Digits data sets. The sum of square error, ϕ, between all the objects and their closest centres, is chosen as the evaluation metric. Obviously, the smaller the ϕ value the higher quality the clustering result achieves.

The experiments are conducted on a Pentium 4 machine with 2.66 GHz CPU and 2 GB RAM, running Windows XP. The algorithms are implemented using Matlab 7.

Table 3
Experimental results on five data sets.

Data set	K-means	CLARANS	VSH	CCIA	CSI	kd-tree	HC_AB_K-means	HC_AB_CLARANS	HC_AB_VSH
RandomS	0.32	0.303	0.307	0.265	0.245	0.282	0.223	0.221	0.221
Statlog	0.433	0.427	0.413	0.389	0.361	0.393	0.312	0.298	0.302
Digits	0.804	0.778	0.792	0.698	0.726	0.743	0.561	0.558	0.558
Water	0.219	0.175	0.202	0.144	0.152	0.157	0.112	0.099	0.102
Sponge	0.108	0.046	0.085	0.039	0.041	0.038	0.035	0.029	0.031

4.1. Clustering quality results

We first run each compared K-centre algorithm on these datasets for M' times (M' increases from 2 by step 2), and then we determine the best one which makes ABS and ABP value achieving the best tradeoff as the number of estimated optimal solutions, i.e., the parameter M of HC_AB described in Section 3. After the determination of M, we run K-means, VSH and CLARANS under the HC_AB framework, and denote them as HC_AB_K-means, HC_AB_VSH and HC_AB_CLARANS respectively. For each adapted HC_AB algorithm, we run each original K-centre clustering algorithm for $M-1$ times respectively to generate its corresponding sub-optimal clustering result collection $Z = \{V^1, V^2, \ldots, V^{M-1}\}$, and then we form the AB from Z and refine the initialization space by using the proposed algorithm, last we re-run these K-centre clustering algorithms with the new constructed initialization setting to obtain the final clustering result. Meanwhile, we also perform the experiments with these three traditional K-centre clustering algorithms and three modified K-centre clustering algorithms with the improvement on initialization. We carry out these six clustering algorithms for M times and choose the best one out the clustering results for comparison, to check whether our proposed approach is able to outweigh them in terms of clustering quality. The detailed experimental results are presented in Table 3.

From Table 3, we find that the ϕ values of HC_AB_K-means, HC_AB_VSH and HC_AB_CLARANS are consistently smaller than those of other six algorithms on five data sets, which justifies the capability of our proposed approach dealing with the initialization sensitivity in K-centre clustering, therefore overcoming the drawback of traditional heuristic clustering algorithms. Furthermore, it is shown that the ϕ value of CLARANS algorithm is smaller than those of K-means and VSH, indicating that CLARANS is more robust in handling the initialization problem. Although CLARANS is able to deal with the initialization sensitivity problem in K-centre clustering algorithm via the random restart method, it still has another drawback of missing the better clustering results [19]. The finding of the ϕ values of VSH being smaller than those of K-means on five datasets is probably because that VSH is a noise-insensitive algorithm by using the distribution method of p devices [17]. However, VSH and K-means are both sensitive to the initialization.

Compared to tradition K-centre clustering, CCIA, CSI and kd-tree algorithms, which use different ways to manipulate the initialization sensitivity in K-means algorithm, are able to achieve the improvement in terms of the ϕ values. However, their clustering performances are still much lower than our approaches by up to 25%. As a result, we conclude that our proposed AB based heuristic clustering algorithm is not only able to effectively improve the clustering result quality against the traditional K-centre clustering algorithms, but also consistently outperform other modified K-centre methods with the improvement on initialization.

4.2. Efficiency comparisons and complexity discussions

On the other hand, we also evaluate the efficiency (i.e. time consumption) of each compared clustering algorithm on five datasets. We run these six compared clustering algorithm for M times and keep the average time consumptions as the result. For HC_AB, the whole time cost of HC_AB consists of two parts: the time cost of generating sub-optimal clustering results collection Z, and the time cost of finding AB and re-running K-centre clustering. Note that the time cost for obtaining the $(M-1)$ sub-optimal clustering results in a sequential execution manner would be $(M-1)$ times of the cost for running a round of the tradition K-centre algorithm, such as K-means. Thus it is not fair to compare the adapted HA_AB algorithms with their counter-algorithms in this way. In order to increase the reliability of efficiency comparison, we choose some technical means to reduce the overhead for preparing the sub-optimal results in our experiment. One solution is that we use parallel computing mechanism to generate the clustering results collection Z at first. In order to clearly describe the time cost of HC_AB, we give an example of HC_AB_K-means on RandomS data set. We run K-means algorithm by using $M-1$ threads on RandomS data set to generate the clustering results collection Z simultaneously, and get the average time cost of 9.71 s as for the first step. By adding the time cost 2.67 s of second steps of HC_AB_K-means, we obtain the total time consumption of HC_AB_K-means, i.e., 12.38 s. The time costs of HC_AB_CLARANS and HC_AB_VSH are generated in the same way. We give the time consumption results in the last three column of Table 4.

From Table 4, we can find that the time consumptions of CLARANS, VSH, CCIA, CSI and kd-tree methods are higher than those of K-means on five data sets. For CLARANS and VSH, their computational complexity is $O(N^2)$. The time consumptions of CCIA and CSI depend on the time cost of new initialization constructing process and the searching process. For kd-tree clustering algorithm, the kd-tree of data objects must be constructed at first and its computational complexity is $O(N\log(N))$. The analysis of the time cost of CLARANS, VSH, CCIA, CSI and kd-tree methods conforms to the phenomena shown in Table 4.

From Table 4, we also could see that the time cost of the proposed algorithms is in almost the same rang of that

Table 4
Time consumption of compared clustering algorithms.

Data set	K-means	CLARANS	VSH	CCIA	CSI	kd-tree	HC_AB_K-means	HC_AB_CLARANS	HC_AB_VSH
RandomS	9.72	162.88	123.67	99.02	81.72	169.43	12.38	167.87	130.29
Statlog	1.34	89.65	67.66	45.43	30.75	102.24	16.89	109.47	72.64
Digits	8.65	164.96	138.78	85.64	67.92	178.99	13.67	170.36	144.31
Water	5.44	112.39	100.87	91.29	81.57	156.42	20.12	118.34	110.81
Sponge	0.86	35.75	22.32	37.29	26.37	90.44	6.07	40.77	31.73

of counter-algorithms: K-means, CLARANS and VSH. For example, HC_AB_CLARANS only needs 4.99 s (167.87 s − 162.88 s = 4.99 s) extra time cost over that of CLARANS on RandomS data set. There are two reasons for this phenomenon: (1) the parallel computing mechanism dramatically reduces the time cost of the processors of generating the clustering results collection Z; (2) the initialization derived from AB increases the converging speed of K-centre clustering algorithm. As a result, we can conclude that although the HA_AB algorithms will incur in some extra time for generating the initialization, the convergence of clustering speeds up and the final cluster results are of better quality.

5. Conclusion

Heuristic clustering is sensitive to initializations and is prone to sub-optimal solutions. Due to the strength of AB on improving the performance of heuristic algorithms, many studies have introduced it in heuristic clustering algorithms. In this paper, we have proposed a novel solution to this by devising an Approximate Backbone based K-centre clustering approach. The main strength of the proposed method is the capability of restricting the initialization space around the optimal results by using the Approximate Backbone, and in turn, reducing the impact of initialization and improving the performance of heuristic clustering. Experiments on synthetic and real world data sets in comparison with traditional and modified K-centre clustering algorithms have shown that the proposed approach possesses the capability of improving the quality of clustering and reducing the initialization impact.

References

[1] J. Han, M. Kamber, A.K.H. Tung, Spatial clustering methods in data mining: A survey, Geographic Data Mining and Knowledge Discovery (2001) 1–29.
[2] X. Rui, D. Wunsch, Survey of clustering algorithms, IEEE Transactions on Neural Network 16 (03) (2005) 645–678.
[3] J.G. Sun, J. Liu, L.Y. Zhao, Clustering algorithms research, Journal of Software 19 (1) (2008) 48–61.
[4] J. Brendan, D.F. Deblert, Clustering by passing messages between data points, Science 315 (2) (2007) 972–976.
[5] P. Drineas, R. Frieze, S. Vempala, et al., Clustering large graphs via singular value decomposition, Machine Learning 56 (1–3) (2004) 9–33.
[6] Z.H. Zhou, W. Tang, Cluster ensemble, Knowledge-Based System 19 (2006) 77–83.
[7] B. Piotr, D.G. Bhaskar, M.Y. Kao, et al., On constructing an optimal consensus clustering from multiple clusterings, Information Processing Letters 104 (2007) 137–145.
[8] K.D. Boese, Cost versus distance in the travelling salesman problem, Technical report CSD-950018, 1995.
[9] P. Merz, B. Freisleben, Fitness landscapes and memetic algorithms and greedy operators for graph bi-partitioning, Evolutionary Computation 8 (1) (2000) 61–91.
[10] C.R. Reeves, Landscapes, operators and heuristic search, Annals of Operation Research 86 (1) (1999) 473–490.
[11] S. Kirkpatrick, G. Toulouse, Configuration space analysis of travelling salesman problems, Journal de Physique 46 (1985) 1277–1292.
[12] W. Zhang, Phase transitions and backbones of 3-sat and maximum 3-sat, in: Processings of the 7th International Conference and Practice of Constraint Programming, 2001, pp. 152–167.
[13] S. Climer, W. Zhang, Searching for backbones and fat: A limit crossing approach with applications, in: Processings of the AAAI2002, 2002, pp. 707–712.
[14] P. Zou, Z.H. Zhou, G.L. Chen, Approximate backbone guided fast ant algorithm to QAP, Journal of Software 16 (10) (2005) 1691–1698.
[15] H. Jiang, X.C. Zhang, G.L. Chen, Exclusive overall optimal solution of graph bipartition problem and backbone compute complexity, Chinese Science Bulletin 52 (17) (2007) 2077–2081.
[16] H. Jiang, X.C. Zhang, G.L. Chen, Backbone analysis and algorithm design of QAP, Chinese Science 38 (01) (2008) 1–14.
[17] J.B. Michael, F.K. Hans, Comment on "Clustering by passing messages between data points", Science 319 (2008) 726c–727c.
[18] Y. Zong, H. Jiang, M.C. Li, Approximate backbone guided reduction algorithm for clustering, Journal of Electronics and Information Technology 31 (12) (2009) 2953–2957.
[19] J. Macqueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, 1976, pp. 281–297.
[20] R. Ng, J. Han, CLARANS: A method for clustering objects for spatial data mining, IEEE Transactions on Knowledge, Data Engineering 14 (5) (2002) 1003–1016.
[21] S.S. Khan, A. Ahmad, Cluster centre initialization algorithm for K-means clustering, Pattern Recognition Letters 25 (11) (2004) 1293–1302.
[22] P. Kang, S.Z. Cho, K-means clustering seeds initialization based on centrality, sparsity, and isotropy, in: Proceedings of the 10th International Conference on Intelligent Data Engineering and Automated Learning, 2009, pp. 109–117.
[23] S.J. Redmond, C. Heneghan, A method for initialising the K-means clustering algorithm using kd-tree, Pattern Recognition Letters 28 (8) (2007) 965–973.