Coordination of Cluster Ensembles via Exact Methods

Ioannis T. Christou, *Member*, *IEEE*

Abstract—We present a novel optimization-based method for the combination of cluster ensembles for the class of problems with intracluster criteria, such as Minimum-Sum-of-Squares-Clustering (MSSC). We propose a simple and efficient algorithm—called EXAMCE—for this class of problems that is inspired from a Set-Partitioning formulation of the original clustering problem. We prove some theoretical properties of the solutions produced by our algorithm, and in particular that, under general assumptions, though the algorithm recombines solution fragments so as to find the solution of a Set-Covering relaxation of the original formulation, it is guaranteed to find better solutions than the ones in the ensemble. For the MSSC problem in particular, a prototype implementation of our algorithm found a new better solution than the previously best known for 21 of the test instances of the 40-instance TSPLIB benchmark data sets used in [1], [2], and [3], and found a worse-quality solution than the best known only five times. For other published benchmark data sets where the optimal MSSC solution is known, we match them. The algorithm is particularly effective when the number of clusters is large, in which case it is able to escape the local minima found by K-means type algorithms by recombining the solutions in a Set-Covering context. We also establish the *stability* of the algorithm with extensive computational experiments, by showing that multiple runs of EXAMCE for the same clustering problem instance produce high-quality solutions whose Adjusted Rand Index is consistently above 0.95. Finally, in experiments utilizing external criteria to compute the validity of clustering, EXAMCE is capable of producing high-quality results that are comparable in quality to those of the best known clustering algorithms.

Index Terms—Clustering, machine learning, constrained optimization, combinatorial algorithms.

1 Introduction

Cluster ensembles [4], [5], [6], [7], [8] have emerged as a recent offspring of the classifier ensemble research area. The basic idea is to combine the results of a number of possibly weak clustering algorithms to produce a final clustering of a data set that is better than each individual algorithm alone can produce. The idea is essentially borrowed from the basic premise of pattern classifier ensembles [9], [10], where several techniques such as Boosting and Bagging have been shown to substantially improve the performance of the individual base classifiers used in the ensemble. This basic premise was shown to hold in the context of cluster ensembles in a number of experiments conducted during the past few years. Strehl's work [11] showed that by combining a few weak clustering results in one final clustering using the Mutual Information Criterion, significantly higher accuracy could be achieved on synthetic as well as real data sets. Strehl also showed that clusterings produced by base clusterers working on a subspace of the original feature space (Feature-Distributed Clustering, or FDC method), or even by base clusterers working with a subset of the original data set (Object-Distributed Clustering, or ODC method), can be efficiently combined to produce highly accurate final results guided by the (Normalized) Mutual Information Criterion.

In [12], it is shown how several weak clustering results can be combined using Mutual Quadratic Information objective criteria to obtain clusters that exhibit better classification accuracy using external criteria, i.e., data point class labels that are unavailable throughout the clustering process. Base partitions are computed via either K-Means runs, or via random projections in 1D lines, or even via random hyperplane splits. Then, these base partitions are combined via a consensus function based on mixture models of consensus or on information-theoretic consensus models.

More recently, a number of UCI data sets [13] were used in an evaluation of the clustering accuracy produced by ensembles of up to 2,500 individual K-Means runs combined in a similarity matrix and then clustered via Single-Linkage criteria [14]. Single Linkage is a well-known bottom-up hierarchical algorithm that works by initially assigning each data point to its own cluster and then, in each iteration, merging the two clusters that are closest (or most similar) to each other in the sense that the two points that belong to different clusters and are closest to each other belong in these two clusters. The process continues until there is only one cluster containing all data points, forming a tree hierarchy of clusters. Usually, the final clustering is chosen by selecting from this tree a point to cut which shows a spike in some internal objective criterion curve. The study evaluated the relation between cluster ensemble accuracy and stability, where classification accuracy was evaluated as the Adjusted Rand Index of the final clustering and the externally given labels. The results were interesting in that they reveal that different data sets exhibit completely

• *The author is with Athens Information Technology, 19Km. Markopoulou Ave., PO Box 68, Paiania 19002, Greece, and the Information Networking Institute, Carnegie-Mellon University, Pittsburgh, PA. E-mail: ichr@ait.edu.gr.*

Manuscript received 5 Feb. 2009; revised 3 Nov. 2009; accepted 15 Feb. 2010; published online 16 Apr. 2010.
Recommended for acceptance by L. Bottou.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2009-02-0088.
Digital Object Identifier no. 10.1109/TPAMI.2010.85.

different relations between the clustering's stability and accuracy, which essentially means that stability is not an indication of accuracy, as is sometimes assumed. In [15], the author presents a more mathematical treatment of clustering comparison, analyzes the relationship between clustering stability and accuracy, and provides an axiomatic characterization of the (unadjusted) Rand Index criterion.

In an optimization context of combining cluster ensembles results, Singh et al. [16] provide an optimization-based framework for the formation of the final clusters so as to maximize consent and minimize dissent of the weak partitions simultaneously. A suitable relaxation of the Nonlinear Binary Program they formulate initially results in a 0-1 Semidefinite Programming problem (SDP); this problem is then further relaxed and transformed to give a final SDP approximation which is solvable in polynomial time. They then use a rounding scheme based on a winner-take-all approach to produce a final feasible clustering. Their results indicate that their scheme performs better than the base clustering solutions used in terms of classification for the majority of their test cases. Contrary to the work presented in [12], their experimental results combine only a few but sophisticated base clustering algorithms.

In this work, we are interested in the combination of weak clustering algorithms in order to produce a better clustering in terms of the objective function that the original weak base methods aim to minimize. The main contributions of this work are the following:

1. The method to combine clustering results is not based on consent among the partial solutions, as is often the case in the literature, but rather attempts to directly optimize the original objective function of the problem at hand.
2. The algorithm is iterative and expands during its iterations the clustering base which is used to form the final clustering solution.
3. The algorithm carries guarantees on its improved accuracy against the base solutions used for a large class of problems.

From point 1 above, it follows that our algorithm is mainly a novel optimization method for optimizing the objective function modeled in a wide range of clustering problems utilizing internal criteria, including Minimum-Sum-of-Squares Clustering [17], Structural-Entropy-Minimization-based Clustering [18], p-Median Clustering [19], and others, as will be made clear in the next section.

In order to combine base clustering solutions using the same objective function as the one that the base algorithms use, we are led to a Set-Partitioning formulation with a side constraint, where the sets to choose from are the clusters produced by each weak method, with the same objective. We show both theoretically and experimentally how to combine solutions of weak base clustering methods for clustering problems with intracluster criteria that obey the *Monotone Clustering Property* (i.e., criteria where adding a data point to a cluster will only increase the cluster's cost [17]) in order to overcome local minima that trap the base clustering methods into low-quality regions. Indeed, the experimental results we present indicate that especially for large values of k—the number of clusters sought—the solution quality found by the proposed method EXAct Method-based Cluster Ensembles (EXAMCEs) can be more than an order of magnitude better than the best solution found by local search methods such as K-Means, and it usually outperforms even the best known methods for the MSSC problem on published benchmark data sets.

The remainder of this paper is organized as follows: In Section 2, we define the class of problems that we are interested in, namely, Intracluster criteria-based clustering problems. We provide a Set-Partitioning formulation of the problems in this class, and based on that formulation, in Section 3, we define the EXAMCE algorithm and establish its theoretical properties. In Section 4, we present the computational results from a large number of experiments we performed on many diverse data sets and compare them directly with previously published results. Finally, we conclude in Section 5 with a list of future directions for this research.

2 Intracluster Criteria-Based Clustering

2.1 Definitions

Consider a finite data set of n d-dimensional data points $S = \{s_1, \ldots, s_n\} \subseteq R^d$.

Definition 1. *We define as* Intracluster criterion-based clustering *(*IC^3*) any clustering method that attempts to determine the optimal partition C of the set S into* disjoint *clusters $C_1, \ldots, C_k \subset S$ such that $\bigcup_{i=1}^{k} C_i = S$ that minimize an objective function of the form $f(C) = \sum_{i=1}^{k} c(C_i)$, where $c(.)$ is some intracluster cost function.*

In other words, the clustering objective can be decomposed in the costs of each cluster, and the cost of each cluster depends only on the cluster itself and is independent of the way the rest of the data set is partitioned. In addition, we have the following definition:

Definition 2. *A clustering criterion satisfies the* Monotone Clustering Property *(MCP) when any two clusters $C_i, C_j \subset S$ satisfy $C_i \subseteq C_j \Rightarrow c(C_i) \leq c(C_j)$.*

As a first example, we notice that the famous MSSC clustering problem is an intracluster criterion-based clustering problem that obeys the MCP. Membership in the intracluster criterion-based clustering problem class follows immediately from the definition of MSSC:

$$\min_{C} . \quad \sum_{i=1}^{k} \sum_{s_l \in C_i} \|s_l - \bar{s}_i\|^2$$

$$s.t.$$

$$\bar{s}_i = \frac{\sum_{l:s_l \in C_i} s_l}{|C_i|} \qquad i = 1, \ldots k,$$

$$\bigcup_{i=1}^{k} C_i = S$$

$$C_i \cap C_j = 0. \quad \forall i \neq j$$

To see that MSSC also obeys the MCP, simply observe that given a finite set of points $X = \{x_1, \ldots x_m\}$, the vector d that minimizes c, the sum of squares of the distances of each

point in X from d, is the mean vector μ of the set X. Then, without loss of generality, consider the set $X' = \{x_1, \ldots, x_{m-1}\}$ whose cost $c' = \sum_{i=1}^{m-1} \|x_i - \mu'\|^2$ is minimized for $\mu' = \frac{1}{m-1}\sum_{i=1}^{m-1} x_i$. Then, $c' \leq \sum_{i=1}^{m-1} \|x_i - \mu\|^2 \leq \sum_{i=1}^{m} \|x_i - \mu\|^2 = c$. This proves the Monotone Clustering Property for the MSSC problem.

A second example of an IC^3 clustering problem that also satisfies the MCP is the minimum-entropy clustering criterion for all values of the externally given constant a *less than 1* [18]. The minimum-entropy clustering criterion is defined as follows for a data set S:

$$\min_{C^k} . \quad J = \begin{cases} 1 - \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k} p^a(c_j \mid s_i), & a > 1, \\ -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k} p(c_j \mid s_i)\ln(p(c_j \mid s_i)), & a = 1, \\ \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{k} p^a(c_j \mid s_i) - 1, & 0 < a < 1, \end{cases}$$

where C^k indicates the set of all possible partitions of the data set among k clusters, and $p(c_j \mid s_i)$ is estimated to be the ratio between the number of samples assigned to cluster c_j in a predefined region (e.g., small hypersphere) of s_i and the number of all samples in that same region. The objective function decomposes into a sum of k terms, each of which is independent of the distribution among clusters of the points that do not belong to the cluster described by each term. Indeed, for $a < 1$, the objective function can be written as $\sum_{j=1}^{k}\sum_{i=1}^{n} \frac{1}{n} p^a(c_j \mid s_i)$ after dropping the constant -1. Obviously, $0 \leq p(c_j \mid s) \leq 1$ for all data points s, so we have that $\frac{1}{n} p^a(c_j \mid s_i) \geq 0$ for all data points. Now, given that $p(c_j \mid s_i)$ can only increase if a data point is added to the cluster c_j, clearly we have that if a cluster is a strict subset of another cluster, then the cost associated with the first cluster is less than or equal to the cost associated with the super set cluster. Parenthetically, notice that the minimum-entropy clustering criterion belongs to the IC^3 for all values of a for which it is defined but for $a \geq 1$ does not satisfy the MCP.

Similarly to MSSC, for the *p-Median problem* [19], [20] (central to location theory) it is not hard to verify that it is a clustering problem that belongs to the IC^3 class *and* satisfies the MCP. A fourth example of clustering criteria that belong to the IC^3 class is the Bayesian Information Criterion (BIC) used in evaluating clustering solutions in X-Means [21]. There are many more clustering criteria that belong to the IC^3 class, so our findings are useful for a large number of clustering problems.

2.2 Set-Partitioning Formulation of the IC^3 Problem

In the optimization literature, a Set-Partitioning formulation has been proposed before [22] in attempts to design algorithms for the *exact* solution of the MSSC problem. More generally, however, the whole class of Intracluster criterion-based clustering algorithms can be formulated as the following Set-Partitioning problem with a side constraint on the total number of clusters allowed in the final solution:

$$(IC^3) \quad \min_{x} . \quad \sum_{i=1}^{N} c_i x_i$$

$$s.t.$$

$$Ax = e,$$

$$x^T e = k,$$

$$x \in \mathrm{B}^N$$

where $A = [a_1 \ldots a_N]$ is a matrix of dimensions $n \times N$, where $N = \sum_{i=1}^{n} \binom{n}{i} = 2^n - 1$ is the total number of subsets of the data set S, the columns of A are the indicator vectors of each partition of the data set S, and c is a vector of N elements whose ith element corresponds to the cost of the ith cluster in the matrix A, $c_i = c(a_i)$; B is the binary set $\{0, 1\}$, e is the vector of ones in $\mathbb{R}^N$, and $x \in \mathrm{B}^N$ is the solution vector, so that if the ith component of x is set to 1, the cluster represented by the ith column of A belongs to the optimal clustering.

Notice that (IC^3), being free to select from all possible subsets of the data set S, remains the same if we replace the equality constraints with the inequality constraints $Ax \geq e$, $x^T e \leq k$, $x \in \mathrm{B}^N$ (see, for example, [23]). This is because in the optimal solution, the inequalities would have to be active (i.e., be satisfied as equalities). To see this, observe that if any row in the constraint $Ax \geq e$ is not satisfied as an equality, then the data point represented by the row will be present in more than one cluster (say clusters j_1 and j_2), and one can lower the objective value by simply selecting the column from matrix A that contains all points of j_1 except for the point corresponding to that row (which is, of course, guaranteed to exist since A contains all possible clusters). But then, the optimal solution is not optimal, a contradiction. However, *when the matrix A in the formulation of the problem is restricted and does not contain all possible subsets of the set S, the problem is no longer the same if we replace the equality constraints with the inequality constraints* and EXAMCE makes heavy use of this fact.

Of course, even for modest values of n, the storage of the matrix A in the formulation of IC^3 becomes practically impossible. Nevertheless, this formulation of the problem provides useful starting points for the design of heuristic algorithms for its solution. The problem is cast as a Set-Partitioning problem with a side constraint, which is, in turn, a Linear Integer Program with 0-1 variables. Usually, to solve such problems, the integrality constraints of the variables are relaxed, and the resulting continuous solution that can be computed in polynomial time yields a lower bound on the original problem (see [24]). Branch-and-bound (B&B) type algorithms [24], [23] then branch on variables that have not been fixed yet, so as to explore the problem that results from fixing some variables to zero or one. For problems with extremely large or possibly infinite number of columns (and thus, variables), column generation techniques [25] provide a robust method for solving such Linear Programs (LPs).

In light, however, of the original context of combining weak clustering results in order to yield a superior final clustering, we immediately see that a *different relaxation* of (IC^3) is possible: One can simply restrict the matrix A to contain only the set of clusters produced by a number of base clustering methods—and let c contain the corresponding cost

of each cluster—and solve the resulting restricted problem to optimality using standard exact methods for Set Partitioning. This simple relaxation leads to interesting results, as we discuss in the next sections.

3 The EXAMCE Algorithm

3.1 Problem Formulation

The observations above lead us to a new formulation for the "fusion" part of a cluster ensemble method for data clustering for IC^3 problems that does not rely on consensus models. Instead, our method (EXAMCE) seeks to optimally recombine partial solutions of different initial clustering results so as to produce better feasible solutions to the original problem, which are then further improved via local search heuristics and subsequently fed back to the algorithm as a new clustering result to be considered in the next iteration. These major outer iterations proceed until no further improvement can be made.

The recombination step consists of a search for the *globally optimal solution of a restricted Set-Covering problem with a side constraint* on the number of clusters contained in the final solution. The optimal solution to the set-covering problem is then transformed into a feasible solution of the (IC^3) problem that, under the MCP, it is guaranteed to be at least as good as the best base clustering result used. As mentioned in the previous section, the optimal solution and value to the (IC^3) problem does not change if we relax the first equality constraint to inequality, effectively turning the problem into a Set-Covering problem with a side constraint, but this invariance property does not hold if the matrix A in the formulation is restricted to containing a *strict (and small)* subset of all the possible groups of points from S. Instead, we define the problem (SCP_R) to be a relaxation of the original (IC^3) problem (in its Set-Covering formulation), where the matrix A is restricted to the matrix A_{B} (having q columns) that only contains the groups returned as solutions by the base clusterers. The new restricted problem becomes:

$$(SCP_R)\ \min_x \sum_{i=1}^{q} c([A_B]_i)x_i$$

$$s.t.\ \begin{cases} A_B x \geq e \\ \sum_{i=1}^{q} x_i = k, \\ x_i \in \mathrm{B} \quad i = 1 \ldots q. \end{cases}$$

The optimal solution x^* to the above problem (SCP_R) certainly has no guarantee of being the optimal solution to (IC^3) and, in fact, may not even be a valid solution to it. But if it is not a feasible solution for (IC^3), then x^* can easily be converted to a feasible solution $\hat{x}$ for the original problem. For this purpose, any transformation that removes duplicate appearances of a data point in the solution produced by (SCP_R) will suffice. Notice that solving (SCP_R) to optimality is still an NP-Hard problem, but, in practice, (SCP_R) can often be solved quite easily, whereas solving exactly the (IC^3) problem is currently out of reach for problems with more than a few hundred data points.

3.2 Algorithm Description

After solving the problem (SCP_R), and, if needed, applying any duplicates removal transformation, we have a valid clustering solution $\hat{x}$ (and when the clustering criterion satisfies the MCP, we have a guarantee that the transformed solution's quality is at least as good as the best solution found by any of the base clusterers, see Theorem 1), but it may not be locally optimal. Therefore, we may apply *any* hill-climbing algorithm for the original problem—e.g., for MSSC problems, the K-Means algorithm—with $\hat{x}$ as the initial starting point and further improve on the solution quality. We may then add this new clustering back into our base pool of clusters by expanding the matrix A_{B} to include the new clusters contained in the new solution and repeat the process until there is no more improvement to be made. Expansion of the base clusters pool, of course, only occurs when the solution $\hat{x}$ is not at a local minimum, and serves to add columns to the (SCP_R) Set-Covering Problem; most of these columns are likely to be present in the final solution, and it is for this reason that the process described took less than five major iterations to converge in all our experiments.

The EXAMCE algorithm is generically defined in pseudocode in Fig. 1 and is highly customizable with respect to the base clustering algorithms used as well as its postprocessing and search-neighborhood expansion heuristic procedures. We now describe the details of the postprocessing and search-neighborhood expansion heuristics chosen *and they are the same in all our tests.*

3.2.1 Search-Neighborhood Exploration and Exploitation Heuristics

The algorithm template utilizes three different procedures as follows:

1. The procedure `Rm_Dup`(x, S_B, c) removes any duplicates from the solution of the Set-Covering Problem solved in step 2.2 of the algorithm in order to produce a valid clustering solution to the initial problem. First, we record all duplicate points (points that appear in more than one cluster of the set-covering solution) along with the ids of the clusters where they appear. Then, each duplicate data point is removed from all the clusters to which it belongs except the one whose center is closest to it, effectively following a greedy nearest-neighbor approach. Cluster centers are updated after each removal.
2. The procedure `Local(C)` is a local search procedure that starts with C as initial clustering and improves on the clustering cost criterion. For the MSSC criterion, we use the HK-Means algorithm [1], which applies a K-Means run starting with initial clusters the solution C, and is then followed by a single-move steepest descent algorithm (for MSSC tests, K-Means with random restarts also serves as the base clusterer). For Entropy Minimization-based clustering, we run the iterative improvement algorithm (called Algorithm 1) described in [18], which also serves as the base clustering algorithm in this case.
3. The procedure `Expand(C)` aims to expand the set of available solutions to consider in step 2.2 by producing a set of clusters that are perturbations of the input cluster C. A trivial procedure may simply return the empty set. In our experiments, we found it beneficial to use a simple heuristic whereby we

EXAMCE

Inputs:
S – a finite collection of points in R^d
k – the number of clusters to partition the data set
c(.) – a cost function $c:2^S \to R$ that obeys the Monotone Clustering Property
B – a set of base clustering algorithms that produce k disjoint clusters of the set S.
Rm_Dup(.) – Any procedure that removes duplicates in the clusters of a clustering solution in order to produce a feasible clustering result.
Expand(C) – a function $Expand:2^S \to 2^{2^S}$ that is a heuristic procedure that expands the set of solutions by taking as input a cluster C and returns as result a set of sets that are "neighbors" of C.
Local(C) – a function $Local:2^{2^S} \to 2^{2^S}$ that implements any local search algorithm for improving the cost function c from an initial starting solution.

Output: a partitioning P of the data set S among k disjoint clusters that is locally optimal with respect to the cost function c(.)

1. Apply the base clustering algorithms in B to produce an initial set S_B of clusters.

2. Repeat steps 2.0-2.7 until no further improvement in clustering cost is made:
2.0 Set $N=|S_B|$.
2.1. Set $(A_B)_{|S| \times N}$ to be the matrix whose columns are the membership indicator vectors of the clusters of S_B.
2.2. Solve the following Set-Covering problem with side-constraint:

$$(SCP_R) \quad \min_x \sum_{i=1}^{N} c([A_B]_i)x_i$$

$$s.t. \begin{cases} A_B x \geq e \\ \sum_{i=1}^{N} x_i = k \\ x_i \in \{0,1\} \quad i = 1 \ldots N \end{cases}$$

2.3. Call Rm_Dup(x,S_B,c) to remove any duplicates from the clusters selected by x, which in turn produces a new set of k clusters C′ that are a feasible solution to the original problem.
2.4. Call Local(C′), to produce a new clustering solution C′′=Local(C′) and evaluate the new cost.
2.5. Set C′′′ = C′ U C′′.

2.6. Call Expand(C) for each C in C′′′ to produce a set $C^4 = \bigcup_{C \in C'''} Expand(C)$.

2.7. Add the set $C^5 = C^4 \cup C'''$ to the set S_B.

3. Return the partitioning solution C′′.

4.End

Fig. 1. Algorithm EXAMCE.

choose the τ closest neighbors of the center of C that are not in C, as well as the τ members of C that are farthest from the center of C, with τ being a small user-defined parameter (set after experimentation to 10). We then create and return a sequence of 2τ new clusters $N_1^+ \subset N_2^+ \subset \ldots N_\tau^+, \quad N_1^- \supset \cdots \supset N_\tau^-$, where each of the N_i^+ clusters contains all points in C plus up to the ith closest nonmember neighbor of C, and each of the N_i^- clusters contains all points in C except up to the ith farthest member of C. The average improvement in objective function value due to this procedure was close to 0.5 percent.

3.3 Some Properties of the Algorithm

As a first property of the algorithm, observe that it can easily handle *constraints on the maximum size of each final cluster* [26] by simply ensuring in step 1 that no cluster whose size falls outside the required bounds enters the matrix A_B, and by appropriately modifying the heuristic procedures for steps 2.4 and 2.6 so that no cluster with infeasible size enters the matrix A_B subsequently.

Next, we establish the finite termination of the algorithm.

Lemma 1. *The EXAMCE algorithm terminates after a finite number of steps.*

Proof. The cost function is bounded from below since there are finitely many partitions of the data set S. Since the algorithm stops as soon as it fails to strictly improve on the objective function, the algorithm has to stop after a finite number of steps. □

A much more interesting property of EXAMCE is that despite the fact that it solves a Set-Covering problem instead of Set Partitioning, the transformation of the Set-Covering solution is guaranteed to provide a clustering solution that is at least as good as the best solution that solving the problem as Set Partitioning could provide.

Theorem 1. *For an* IC^3 *problem with a clustering criterion that satisfies the Monotone Clustering Property, the EXAMCE algorithm will find a solution at least as good as the optimal clustering solution of the restricted version of* IC^3 *that considers only the columns produced by the base algorithms B of EXAMCE.*

Proof. Consider the first iteration of the algorithm. Let x^* be the optimal solution of the Set-Partitioning problem:

$$(SPP) \quad \min_x \sum_{i=1}^{|S_B|} c([A_B]_i)x_i$$

$$s.t. \left\{ \begin{array}{c} A_B x = e, \\ x^T e = k, \\ x_i \in \{0,1\} \quad i = 1 \ldots |S_B| \end{array} \right\}.$$

This solution is also a feasible solution of the problem (SCP_R), and therefore, the optimal solution x_c^* of (SCP_R) is always as good as or better than the optimal solution of (SPP). Now, if the optimal solution of (SCP_R) is also a feasible solution of (SPP), then the theorem holds. Otherwise, EXAMCE will apply the duplicate-removing procedure to remove duplicates from the solution x_c^*, which will produce a new clustering solution $\hat{x}^*$ which will not be a feasible solution to the (SPP) problem since it contains clusters not contained in A_B. This solution, however, is a feasible solution of IC^3 and its cost is $\sum_{C \in \hat{x}^*} c(C) \leq \sum_{C \in x_c^*} c(C)$ because it only differs from x_c^* in the clusters from which it has removed duplicate data points, and so, by the Monotone Clustering Property, its cost is lower than the cost of x_c^*. Subsequent iterations of the EXAMCE algorithm will only improve upon the solution $\hat{x}^*$ found during the first iteration, so the theorem holds. □.

It is interesting to note that if the solution of (SCP_R) is not a feasible solution for (SPP), then the algorithm will find a strictly better solution than the solution of the (SPP) problem since it will generate new—reduced cost—columns that will participate in the intermediate solutions of the problem. So, the algorithm essentially starts with an initial set of partial solutions (clusters), and then recombines them or generates, on the fly, new ones to produce new improved clustering solutions which are then locally improved and later used to expand the "search neighborhood" of partial solutions in the major iteration loop of the algorithm until no more improvement of the clustering criterion is possible.

3.4 Computational Complexity Considerations

The computational speed of the algorithm that was evidenced in the experiments performed is due, to a large extent, to the low integrality gap between the continuous relaxation of (SCP_R) and its integral solution, which, as will be demonstrated later, allows the optimal solution to be found quickly even for problems involving tens of thousands of variables and thousands of constraints. Notice, however, that the EXAMCE algorithm is not polynomial due to the requirement in step 2.2 to solve to optimality a Set-Covering Problem with a side constraint, which is NP-hard. We can obtain a polynomial-time version of EXAMCE by relaxing the integrality constraints of problem (SCP_R) in step 2.2 with the constraint $0 \leq x \leq e$, solving the relaxed Linear Program in polynomial time, and then obtain a suboptimal integral solution by choosing the k columns with the highest values in the optimal solution, and assigning any unassigned points (data points that are not in any of the chosen columns) to the cluster with the closest center. This polynomial-time version of the algorithm, however, does not carry the property described in Theorem 1 since the solution obtained from this heuristic procedure will be a feasible solution for the (IC^3), but it is *not* guaranteed to be the optimal solution to (SCP_R) and therefore may have an objective value bigger than the optimal solution to the (SPP). In practice, however, it turns out that this polynomial-time approximation works quite well, and in the next section, we report on results obtained from running this scheme as well.

4 Computational Results

4.1 Preliminaries

All of the experiments described in this section were performed on a Dell Optiplex 755 equipped with an Intel Core 2 Quad CPU with a clock speed of 2.4 GHz and 2.0 GB of RAM running Microsoft Windows XP. The experiments are divided in two categories. In Section 4.2, we report results from running experiments with clustering criterion, the MSSC criterion. *The purpose of this suite of tests is to compare EXAMCE performance as an optimization algorithm against the best known algorithms for MSSC optimization.* We report, in particular, the results from the TSPLIB benchmark data sets that were originally used in [1] and subsequently compared in [2] and [3]. We also compare EXAMCE with *all known exact method-based solutions* that publish their performance on publicly available data sets, where we also find the optimal solution in all but one case. Overall, the TSPLIB data sets u1060, pcb3038, and rat575, and the UCI data sets iris, soybean-small, and segmentation were used to assess EXAMCE performance on Minimum-Sum-of-Squares Clustering. A synthetic data set ("Gauss100") was created by the author using the same method as the authors of [2] for testing EXAMCE against standard K-Means with random restarts on large data sets whose cluster shape is such that K-Means is expected to behave well. This data set size (10,000 data points in $\mathbb{R}^5$) is large enough to test the scalability of the algorithm in terms of running time.

In Section 4.3, we report results from experiments with Entropy minimization in order to compare the accuracy of the clusters produced by EXAMCE with that of other clustering methods published in the literature. Clustering

TABLE 1
Data Sets Used in the Experiments

Data-set Name	In Library	#Instances	#Dimensions	#Classes	Other Studies Using it	Clustering Criterion
rat-575	TSPLIB	575	2	N/A	[3]	MSSC
u1060	TSPLIB	1060	2	N/A	[1],[2],[3]	MSSC
pcb3038	TSPLIB	3038	2	N/A	[1],[2],[3]	MSSC
Gauss100	synthetic	10000	5	100	-	MSSC, Entropy
Iris	UCI	150	4	3	[1],[22], [28],[14]	MSSC, Entropy
Soybean-small	UCI	47	35	4	[14],[28],[29]	MSSC, Entropy
segmentation	UCI	2310	19	7	[14]	MSSC, Entropy
2d2k	synthetic	1000	2	2	[11]	Entropy
8d5k	synthetic	1000	8	5	[11]	Entropy
thyroid	UCI	215	5	3	[14]	Entropy
wine	UCI	178	13	3	[14]	Entropy
glass	UCI	214	9	6	[14]	Entropy
ionosphere	UCI	351	34	2	[14]	Entropy

accuracy is measured via the Adjacent Rand Index or simply via the error percentage of clustering—using the Hungarian method for optimally assigning cluster indexes to externally given labels, as in [12]—depending on the method the authors of the other studies previously used to measure the accuracy of their algorithms, so as to facilitate direct comparison of results. Overall, the three synthetic benchmark data sets 2d2k and 8d5k from Strehl's thesis and the "Gauss100" data set mentioned above, and the seven UCI data sets iris, thyroid, wine, glass, ionosphere, soybean-small, and segmentation were used for assessing EXAMCE performance on classification accuracy given external criteria, provided by the data sets' class labels. In total, therefore, we tested EXAMCE on 13 distinct data sets, as shown in Table 1.

EXAMCE was written in Java and uses the SCIP [27] Open-Source Mixed-Integer Programming solver for solving the Set-Covering problem of step 2.2 to optimality. In fact, the solver parameters are set so that it stops execution after 300 seconds of elapsed time with the best feasible solution it has found until that time, but in our experiments, this limit was never reached and the solver was always able to finish with the optimal solution.

4.2 MSSC-Related Clustering Results

For Minimum-Sum-of-Squares clustering, we view the clustering problem as a pure optimization problem, and use as base clusterers random initializations of the K-Means algorithm, which fits nicely with the general context of cluster ensembles where the research question is how to use in the most profitable way some fast, albeit weak, cluster methods to produce a superior cluster result. The initial centers of K-Means are chosen randomly from the data set. The final clusters produced by each run of the K-Means algorithm are added to the initial set S_B. We run the K-Means algorithm not only for k clusters, but also with different values of k in the set $\{k-w, k-w+1, \ldots k-1, k+1, \ldots k+w\}$ for a number of times, with w being again user-defined (set in all our experiments to $\lfloor \frac{k}{10} \rfloor$) and enter the final clusters of each run in the matrix A_B. Running K-Means with different values of k (resembling bracket exposure techniques in photography) has been shown to boost performance of cluster ensembles in a number of studies (e.g., [14]). The first experiments we report compare the performance of EXAMCE with the best published results on the two TSPLIB benchmark data sets *u1060* and *pcb3038* [1] that are publicly available for download. In the results shown in Tables 2 and 3, we compare the performance of EXAMCE against five different algorithms that are considered the best known methods for MSSC. The first column in each table denotes the number k of clusters sought. The numbers in columns labeled "J-Means" and "VNS+" are directly copied from [1], whereas the column "GA-QdTree" shows the results reported in [2], where it is mentioned that these results are the best known. J-Means is a local-search heuristic for the MSSC problem that

TABLE 2
Comparison of Results of Best Known Methods for Minimum-Sum-of-Squares Clustering for the u1060 TSPLIB Data Set

k	J-Means	VNS+	GA-QdTree	SS	DA	LPMCE	EXAMCE
10	1.75642E+09	**1.75484E+9**	1.75585E+9	**1.75484E+9**	1.75651E+9	1.75538E+9	**1.75484E+9**
20	8.18953E+08	**7.91794E+8**	**7.91974E+8**	**7.91794E+8**	8.24832E+8	7.91973E+8	**7.91794E+8**
30	5.01415E+08	**4.81251E+8**	4.81551E+8	**4.81251E+8**	5.29609E+8	4.81257E+8	4.81369E+8
50	2.69153E+08	**2.55509E+8**	2.56894E+8	2.56426E+8	3.12432E+8	2.55693E+8	**2.55509E+8**
60	2.05460E+08	**1.97273E+8**	1.98738E+8	1.97376E+8	2.41560E+8	1.97326E+8	**1.97273E+8**
70	1.64344E+08	**1.58450E+8**	1.59908E+8	**1.58450E+8**	2.02692E+8	1.58520E+8	**1.58450E+8**
80	1.33942E+08	**1.28890E+8**	1.29732E+8	1.29315E+8	1.62675E+8	**1.28890E+8**	**1.28890E+8**
90	1.14845E+08	**1.10417E+8**	1.11634E+8	1.10705E+8	1.47533E+8	**1.10417E+8**	**1.10417E+8**
100	1.00098E+08	9.63781E+7	9.85214E+7	9.68604E+7	1.29251E+8	**9.63178E+8**	**9.63178E+7**
110	8.75439E+07	8.48458E+7	8.70672E+7	8.53041E+7	1.17012E+8	8.48496E+8	**8.48396E+7**
120	7.96065E+07	7.55997E+7	7.66384E+7	7.65544E+7	1.10759E+8	**7.55365E+7**	7.55537E+7
130	7.22357E+07	**6.75542E+7**	6.93417E+7	6.84660E+7	1.03051E+8	**6.75542E+7**	**6.75542E+7**
140	6.43549E+07	6.11216E+7	6.27243E+7	6.16235E+7	9.52545E+7	6.11419E+7	**6.11195E+7**
150	5.88170E+07	5.59256E+7	5.77786E+7	5.62115E+7	9.05250E+7	**5.59081E+7**	**5.59081E+7**

always places the center of a cluster in one of the data points and searches for the best move of a cluster center among the data points to be clustered. At the end of every iteration, an HK-Means type algorithm improves upon the solution found that locates centers to data points. Variable Neighborhood Search (VNS) is a heuristic algorithm for solving combinatorial problems by exploring solutions in the neighborhood of a current solution and occasionally moving away from the current neighborhood when improvement seems to stall on successive iterations; the "jump" neighborhood moves further and further away—in a random manner—from the current solution when improvement is not sufficient. The column "SS" shows the published results from a Scatter-Search approach [3] on these testbeds. Scatter Search is a recent addition to Evolutionary Algorithms (EAs) that have a common trait with EXAMCE, namely, that of recombining good solutions to drive a local search. However, Scatter Search borrows essentially from Genetic Algorithms in that pairs of good solutions from a reference set are used to build new solutions in a path linking the ones in the pair. The column "DA" shows the results obtained by running a Java implementation of the Deterministic Annealing method for clustering data [29], where we set the annealing rate α to 0.91, and the T_{min} parameter to 10. The column "LPMCE" shows the value obtained by running the polynomial-time LP-based approximation to solving step 2.2 of EXAMCE, as mentioned in Section 3.4. The final column "EXAMCE" obviously shows the best clustering result we obtained from a single run of EXAMCE as specified here. The two tables are summarized graphically in Fig. 2, where we plot the gap of EXAMCE from the *previously* best known value. The gap (a percentage value) between a solution α and a solution β is defined as $Gap(\%) = 100 \times (a - \beta)/\beta$. Notice that for the pcb3038 data set and for $k = 500$, EXAMCE finds a solution that is almost 2 percent better than the previously best known value.

In Table 4, we compare EXAMCE performance against VNS, Scatter Search, and DA on the smaller TSPLIB data set rat-575 containing 575 points in 2D, as well as against the polynomial-time LP-based approximation to solving step 2.2 of EXAMCE as mentioned in Section 3.4 that is shown in column LPMCE. This is the third and last benchmark test used in the study of the performance of Scatter Search conducted in [3]. The results again show that EXAMCE outperforms both VNS and Scatter Search in that in 7 out of 12 total cases, EXAMCE finds strictly better quality solutions than the other two methods; in three cases, it is on a par with the other two solutions and in only two cases does it find a slightly worse solution than the other

TABLE 3
Comparison of Results of Best Known Methods for Minimum-Sum-of-Squares Clustering for the pcb3038 TSPLIB Data Set

k	J-Means	VNS+	GA-QdTree	SS	DA	LPMCE	EXAMCE
10	5.63500E+08	**5.60251E+8**	**5.60251E+8**	**5.60251E+8**	5.75479E+8	**5.60251E+8**	**5.60251E+08**
20	2.66945E+08	**2.66812E+8**	2.66841E+8	**2.66812E+8**	2.67342E+8	2.67045E+8	2.66858E+08
30	1.76494E+08	1.75598E+8	1.75574E+8	1.75598E+8	1.80780E+8	1.76543E+8	**1.75574E+08**
40	1.28263E+08	1.2607E+8	1.25326E+8	**1.24961E+8**	1.35776E+8	1.27135E+8	**1.24961E+08**
50	1.00606E+08	9.89439E+7	9.86408E+7	9.83401E+7	1.07853E+8	1.01160E+8	**9.82754E+07**
100	4.98814E+07	**4.77197E+7**	4.82489E+7	4.78991E+7	5.95101E+7	5.02058E+7	4.77532E+07
150	3.20699E+07	3.05573E+7	3.10387E+7	3.07096E+7	4.08027E+7	3.31395E+7	**3.05191E+07**
200	2.31570E+07	2.19186E+7	2.24617E+7	2.23114E+7	2.69697E+7	2.40473E+7	**2.19116E+07**
250	1.76716E+07	1.66603E+7	1.72234E+7	1.71146E+7	2.15262E+7	1.73583E+7	**1.66568E+07**
300	1.42153E+07	1.33540E+7	1.38574E+7	1.36574E+7	1.76317E+7	1.36672E+7	**1.32764E+07**
350	1.18071E+07	1.10979E+7	1.15984E+7	1.14076E+7	1.49394E+7	1.12416E+7	**1.10475E+07**
400	1.00790E+07	9.41168E+6	9.88791E+6	9.71048E+6	1.22896E+7	9.58653E+6	**9.39826E+06**
450	8.77182E+06	8.22641E+6	8.57850E+6	8.37388E+6	1.13425E+7	8.14641E+6	**8.13033E+06**
500	7.71113E+06	7.23506E+6	7.54327E+6	7.33810E+6	1.02445E+7	7.12999E+6	**7.11365E+06**

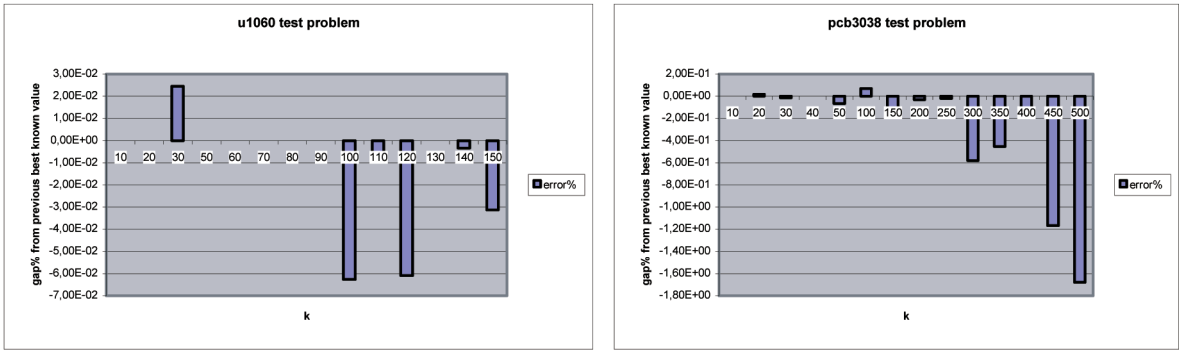


Fig. 2. Percentage gap of EXAMCE from previously best known MSSC value for the u1060 and pcb3038 data sets.

methods. Clearly, DA does not perform as well as the other methods. The column labeled "VNS-HK" in the table shows the performance of the VNS algorithm (with HK-Means as local search algorithm) implemented by Pacheco [3].

Regarding running times of EXAMCE, in Table 5, we compare the minimum and maximum running times required to run EXAMCE and LPMCE on the three benchmarks rat-575, u1060, and pcb3038. The algorithms seem to scale reasonably with data set size increase, and even though the system is written in Java and therefore,has some runtime overheads related to the JVM and uses the noncommercial MIP solver SCIP, the running times are still very reasonable. The reason that LPMCE ran slower than EXAMCE when $k = 500$ is that the LP relaxation resulted in a large number of fractional-valued solutions and the heuristic procedure for obtaining a valid clustering solution then resulted in a large number of iterations of steps 2.0-2.7 of the algorithm. Finally, it is worth mentioning that

TABLE 4
Comparison of Results of Variable-Neighborhood Search, Scatter Search, and EXAMCE Methods for Minimum-Sum-of-Squares Clustering for the rat-575 TSPLIB Data Set

K	VNS-HK	SS	DA	LPMCE	EXAMCE
5	**2497887.4**	**2497887.4**	2528551.3	**2497887.4**	**2497887.4**
10	**1110033.9**	**1110033.9**	1112176.7	**1110033.9**	**1110033.9**
15	**728530.3**	**728530.3**	770329.7	729070.2	728563.3
20	**531913.5**	**531913.5**	576318.8	531939.6	**531913.5**
30	348823.5	**348563.4**	370158.1	359012.3	348880.0
40	255201.7	254968.5	281942.3	268503.6	**254906.8**
50	197830.0	196087.8	224660.5	207757.4	**195650.1**
60	156756.0	156611.9	171869.9	166650.4	**156576.4**
70	129276.6	128863.0	146548.3	129734.6	**128726.1**
80	109963.0	110066.1	122778.7	113543.8	**109699**
90	94828.3	94858.4	113127.1	94872.9	**94453.7**
100	82735.2	82551.2	99571.1	**82442.2**	**82442.2**

TABLE 5
Minimum and Maximum Running Time for LPMCE & EXAMCE Methods for Minimum-Sum-of-Squares Clustering for the TSPLIB Data Sets

Data-Set	LPMCE		EXAMCE	
	Min. Run Time	**Max. Run Time**	**Min. Run Time**	**Max. Run Time**
rat-575	4 secs (k=5)	23 secs (k=70)	4 secs (k=10)	227 secs (k=30)
u1060	36 secs (k=30)	70 secs (k=150)	31 secs (k=60)	157 secs (k=10)
pcb3038	34 secs (k=10)	508 secs (k=500)	24 secs (k=10)	301 secs (k=100)

In parentheses is the number of clusters corresponding to the runtime.

running the DA algorithm implementation took more than 12 hours with the annealing schedule mentioned for each of the last three cases of the pcb3038 data set.

As can be clearly seen from the graphs in Fig. 2 and Tables 2, 3, and 4, EXAMCE has found in 21 cases better solutions than the previously best known solutions and was slightly worse only five times. Statistical analysis (Hypothesis Testing for the absolute gap between EXAMCE and the previously best known solution values through the t-test and sign-test as well as signed-rank-test) reveals that with 95 percent confidence level, the null hypothesis that the mean or the median of the absolute gap is zero must be rejected, and the differences in the results of the algorithms are statistically significant. Also note that an application of X-Means [21]—another state-of-the-art extremely fast algorithm for MSSC clustering that has the advantage of automatically computing the number of clusters—on the u1060 data set returns five clusters with a distortion (i.e., MSSC value) of $3.79152\text{E}{+9}$, whereas running EXAMCE on u1060 with $k = 5$ returns a solution with MSSC value of $3.79100\text{E}{+9}$; similarly, running X-Means on the pcb3038

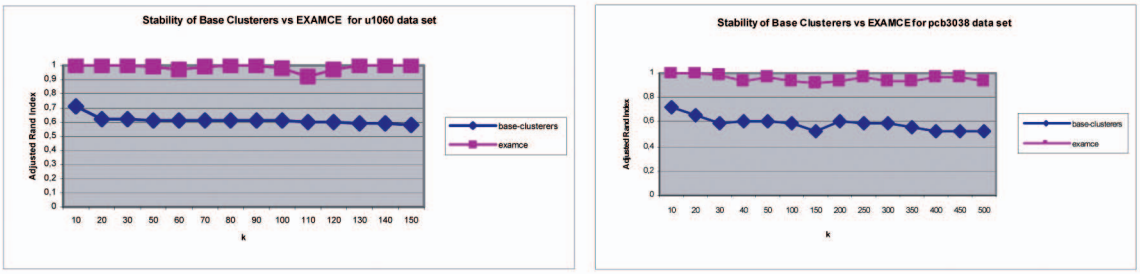


Fig. 3. EXAMCE stability on the TSPLIB u1060 and pcb3038 data set tests.

data set returns 10 clusters with distortion of 5.61709E+8, which is again inferior to the solutions found by VNS+ and EXAMCE, as shown in Table 3.

4.2.1 Stability of the EXAMCE Algorithm

Regarding the stability of the EXAMCE algorithm, Figs. 3 and 4 show the Adjusted Rand Index of the base K-Means clusterers used for the EXAMCE runs described in Tables 2, 3, and 4 and the Adjusted Rand Index of the solutions produced by 10 runs of EXAMCE with different random seeds. The Adjusted Rand Index serves as an excellent stability index for the algorithm that produces the clustering solutions, and the results show clearly (as was to some degree expected) the very stable nature of the EXAMCE algorithm. *Even though the base clusterers have significant variation in them, their recombination yields almost identical solutions, even when the number of clusters sought is very large*. This implies that the random seed used to initialize the base clusterers essentially has no effect on EXAMCE performance.

The next set of experiments is designed to compare EXAMCE performance on larger MSSC problems, both in terms of number of data points as well as number of dimensions.

Because there exist no published performance results on these data sets for J-Means, VNS+, or SS, we compare EXAMCE performance against the HK-Means algorithm with random restarts, as well as our own implementation of Deterministic Annealing (DA). The DA algorithm was run with $\alpha = 0.91$ and $T_{min} = 10$, as in the previous MSSC experiments. Table 6 compares EXAMCE performance against DA and HK-Means on the UCI "*segmentation*" data set with 2,310 data points in 19 dimensions, a higher order dimension data set. Again, the power of recombining partial solutions to compute a much better final solution is shown clearly in the results, where the best solution found by the base clusterers for $k = 300$ is more than 110 percent worse than the EXAMCE solution. Notice that for $k = 200$, the DA algorithm found a better solution than EXAMCE (by 1.3 percent), but for $k = 300$, the DA solution was more than 100 percent worse.

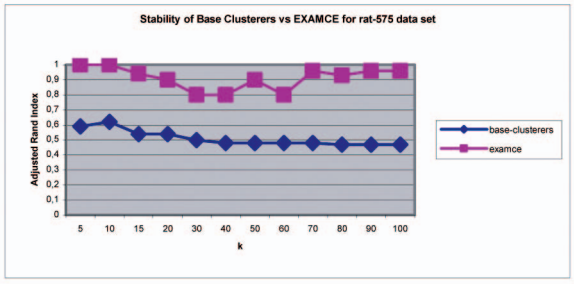


Fig. 4. EXAMCE stability on the TSPLIB rat-575 data set tests.

The next set of experiments was conducted in order to compare EXAMCE performance against the *optimal* MSSC values for the UCI data sets *iris*, *soybean-small* for which the optimal MSSC value is known. The data sets are chosen exactly because the optimal MSSC solution for a variety of k values is known for them. The *iris* (or Fisher's) data set was used in the studies of [22], [28] as well as in [1]. It is a small data set containing 150 instances. The *soybean-small* data set, with only 47 instances in 35 dimensions, is essentially a toy data set and the MSSC problem can be solved to optimality even with brute-force Branch and Bound methods [17].

In Table 7, we provide the best MSSC values found by EXAMCE using the same settings as for the TSPLIB data sets, and compare them with the optimal solutions found by the algorithm described in [22]. (The results published in [22] are slightly different because they use the "original Fisher" data which differ from UCI iris data set in two flower measurements.) As can be seen from the table, *we find the optimal MSSC clustering solutions in all cases*.

The optimal MSSC solutions for the soybean-small data set for the cases $k = 2, 3, 4$ have been published in [28], where the authors used a 0-1 SDP formulation to solve the MSSC problem, and in [30], where the authors experimented with a modified K-means algorithm. With the parameter settings for EXAMCE again the same as before, it obtains the optimal solution in two of the three cases published (for $k = 2, 4$).

In another experiment with MSSC criteria, Fig. 5 shows the effect of adding more base K-Means restarts on the base set B of EXAMCE for the TSPLIB u1060 data set for $k = 100$. In the figure, the results of each column are produced using the base clustering results of the previous column plus some more. It can easily be seen that restarting K-Means many times with different initial parameters has little chance of escaping the many local minima present in problems with large k, as noticed already. The problem has been discussed in detail in [31] and is known as the *central limit catastrophe*. The figure also clearly shows that *significantly better quality solutions can be obtained by combining as few as 10 K-Means clustering results using EXAMCE*. This can perhaps be explained as follows: For large values of k, any single run of the K-Means algorithm is likely to

TABLE 6
The Effect of k on MSSC Algorithms on the Segmentation Data Set

k	DA	Best of 150 K-Means Iterations	EXAMCE MSSC Soln	
			Major Iterations	MSSC Value
100	1404738..1	1799874.9	2	**1370426.2**
150	1083809.5	1351192.9	2	**977548.9**
200	**919088.1**	1199826	2	931294.1
250	815297.9	1034068.1	3	**484173.8**
300	826855.6	874148.9	3	**399982.3**

obtain a few near-optimal clusters in some regions, but is almost certain to "mix" points in many other regions of the space. Combining the clusters of a few K-Means runs in the proposed Set-Covering formulation then can have a big impact in solution quality.

It is also interesting to note that when the number of base partitions increases from 30 to 40, the final solution reported by EXAMCE slightly worsens. This is a rare effect, attributable to the running of step 2.4, the procedure `Local (.)` which produces a local minimum of MSSC starting with the transformed solution of steps 2.2-2.3. The HK-Means run of step 2.4 results in a final better partition when the initial starting point (I_{30}) is the solution of SCP of step 2.2 having combined 30 runs of K-Means, than when the initial starting point (I_{40}) is the solution of SCP having combined 40 runs of K-Means despite the fact that cost $(\mathrm{I}_{30}) > \mathrm{cost}(\mathrm{I}_{40})$.

TABLE 7
MSSC Results for Iris Data Set

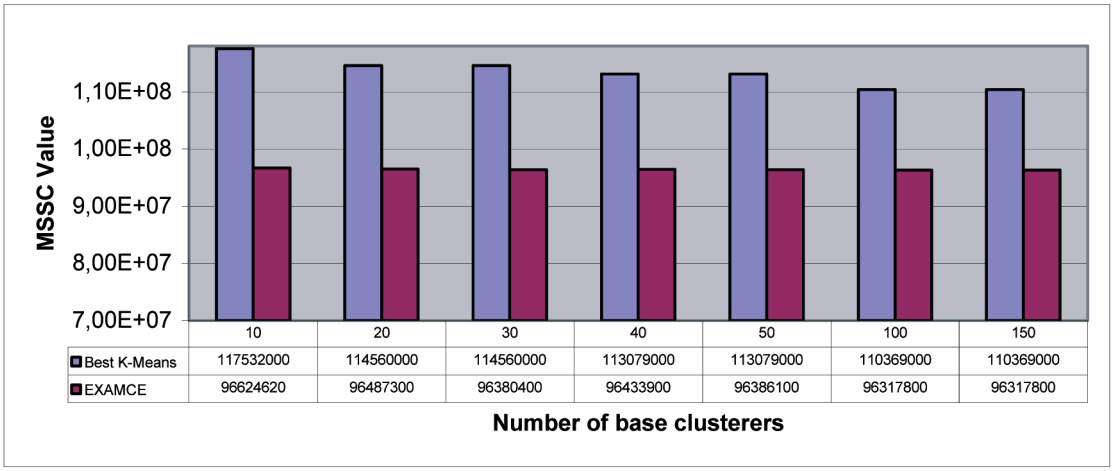
K	Optimal MSSC Soln using the algorithm in [22]	EXAMCE optimality gap%
2	152.368706	0.00
3	78.9408414	0.00
4	57.3178732	0.00
5	46.5355821	0.00
6	38.9309630	0.00
7	34.1892055	0.00
8	29.8799198	0.00
9	27.7654245	0.00
10	25.8133869	0.00

Finally, to check how much better EXAMCE can perform on relatively large-scale problems where the classical K-Means algorithm is expected to behave near-optimally, we clustered the "Gauss100" data set, for different values of k with K-Means (with 150 restarts), and then the resulting clusters were fed to EXAMCE as base clusterer solutions. The results, as shown in Fig. 6, reveal that even when the data set consists of spherical clusters, where K-Means is supposed to excel, it fails to find the optimal clustering solution in terms of MSSC value. The maximum EXAMCE runtime for the "Gauss100" data set was 803 seconds (for the case $\mathrm{k} = 150$), of which less than 146 seconds was spent on solving the Set-Covering Problem to optimality, while the vast majority of time was spent on running the base clusterers.

4.3 Entropy-Minimization Clustering Results

When comparing clustering solutions with clustering algorithms that compute their solution quality using some externally provided criteria, we use Entropy Minimization as the clustering criterion for the problem to be solved. We denote our system `EXAMCE<Entropy>` to indicate that it is configured so that it solves an Entropy-Minimization clustering problem. The reason for choosing Entropy Minimization as the clustering criterion to optimize is that for most real-world applications, where the data do not come from (well separated) Gaussian distributions, Entropy-minimizing algorithms usually provide much better clustering accuracy than K-Means type methods. In this case, we use a single base clustering algorithm, the iterative improvement algorithm (called Algorithm 1) described in [18], with random restarts.

We tested the system on a number of data sets previously used in the literature and compare them with the same criterion published before. We use entropy minimization as clustering criterion as in [18], but with $a = 0.9$. This, of course, has a direct effect on the cost the



	10	20	30	40	50	100	150
Best K-Means	117532000	114560000	114560000	113079000	113079000	110369000	110369000
EXAMCE	96624620	96487300	96380400	96433900	96386100	96317800	96317800

Fig. 5. The effect of the number of base clusterers on EXAMCE for u1060 data set with $k = 100$.

algorithm computes for each cluster it considers. We ran "Algorithm1" 150 times from a random initial partition. All other parameters of the EXAMCE<Entropy> algorithm are as described in Section 3.2.1.

Strehl (www.strehl.com) has made available two of the data sets he experimented with for his PhD thesis on clustering ensembles [11]. These two data sets are synthetic in that they were created as a result of Gaussian processes creating data points in R^2 or R^8, respectively. We use clustering accuracy to compare clustering quality of EXAMCE<Entropy> with Strehl's ensemble methods described in Section 1. Clustering accuracy is computed as one minus the classification error of the clustering solution after an application of the Hungarian algorithm [12] has assigned clusters to class labels. The results are shown in Fig. 7. Obviously, the higher the accuracy, the better is the algorithm that produced it.

In Table 8, we compare clustering accuracy against the known labels of the data (as measured with the Adjusted Rand Index), with the results presented in [14]. We only show results for data sets on which we obtained access to. Interestingly, in all cases, EXAMCE<Entropy> outperforms a much larger ensemble.

In Table 9, we show that comparisons of EXAMCE <Entropy> achieved classification error rate with the best results achieved by five different consensus methods reported in [12], for the *Iris* data set. The settings of EXAMCE are as before. To make the comparison fair, the result of EXAMCE<Entropy> we report is the average of 10 runs of the system with 10 different random number generator seeds, using "Algorithm1" as the scheme for populating the base weak clustering solutions set S_B. However, due to the stability of our algorithm, the

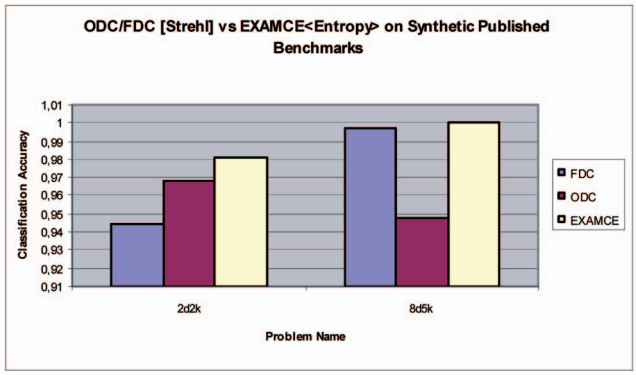


Fig. 7. Comparing clustering accuracy of EXAMCE<Entropy> clustering results and FDC and ODC. EXAMCE minimizes structural entropy.

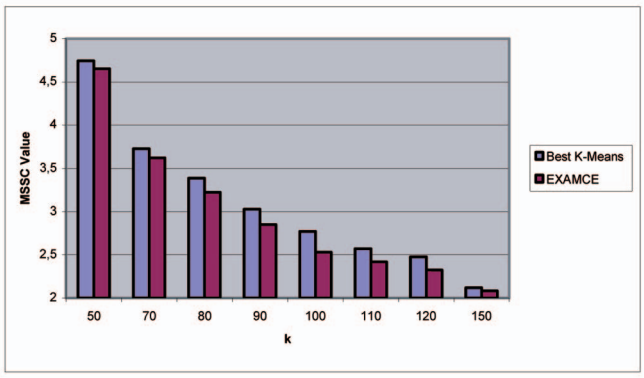


Fig. 6. Comparison of MSSC values found by 150 K-Means runs and EXAMCE for the "Gauss100" data set.

TABLE 8
Comparison of EXAMCE<Entropy> and K-Means Ensembles in Terms of Clustering Accuracy on UCI Data Sets as Measured by Adjusted Rand Index

UCI Data-Set	EXAMCE<Entropy>	K-Means Ensemble reported in [14]
Thyroid	**0.842**	0.594
Wine	**0.731**	0.403
Glass	**0.411**	0.301
Ionosphere	**0.479**	0.296
soybean-small	**1.0**	0.937
Segmentation	**0.519**	0.495

EXAMCE minimizes structural entropy.

TABLE 9
Error Rate Comparisons of Different Clustering Ensemble Methods

Data-set	Best Mean EM [12]	Best Mean QMI [12]	Best Mean CSPA [11]	Best Mean HGPA [11]	Best Mean MCLA [11]	*Mean EXAMCE<Entropy>*
Iris	10.9	10.8	7.7	38.2	10.9	3.3

deviation of the various runs of EXAMCE<Entropy> from the average is essentially insignificant.

Except for the HGPA method, the other methods obtain fairly close accuracies on average.

The last experiment with Entropy minimization was conducted on the synthetic test "Gauss100." We compare the Normalized Mutual Information Criterion value [11] computed for the clustering results of 150 runs of K-Means versus the results of EXAMCE<Entropy> with $\alpha = 0.9$ (where the initial base clusterers are the results of 150 applications of Algorithm1 in [18]). As it turns out (Fig. 8), the proposed scheme outperforms K-Means with restarts in clustering quality judged by external criteria as well, although the differences are not as dramatic.

5 Conclusions and Future Directions

We have presented EXAMCE, a novel iterative algorithm using cluster ensembles for the IC^3 class of clustering problems that directly seeks to optimize the same criterion that the base clustering algorithms attempt to optimize. The algorithm works by solving a Set-Covering problem (with a side constraint) with variables corresponding to the clusters found by the base algorithms, and costs of each variable directly derived from the clustering problem definition. The Set-Covering Problem solution is transformed to a valid clustering solution and, in the case of clustering problems with the Monotone Clustering Property, the solution is guaranteed to be at least as good as the optimal solution of the original restricted Set-Partitioning formulation.

The algorithm is iterative in that it applies local search heuristics to (greedily) improve the Set-Partitioning solution and exploitative in that it adds to the set of partial solutions to be considered in the next major iteration partial solutions in the neighborhood of the currently improved one. The algorithm stops when no improvement in the objective can be made. The total number of major iterations required for convergence of the algorithm is very small, usually between 1 and 2, and never in any experiment conducted more than 5.

We have shown by extensive computational experimentation that the algorithm outperforms all well-known methods for Minimum-Sum-of-Square distances clustering. We have shown that the algorithm is capable of escaping the central limit catastrophe by optimally combining partial solutions found by weak clustering algorithms in a Set-Partitioning reformulation of the problem. This capability of the algorithm is most striking for problems involving large numbers of clusters, which is of particular importance in applications such as fraud detection. We have also shown that the application of the algorithm in an entropy-minimization context for discovering the true structure of the data yields good results even with few initial partial clustering solutions to consider. EXAMCE performs well on revealing clustering structure as measured by the Adjacent Rand Index, even when compared with much larger ensembles based on K-Means.

EXAMCE is an algorithm for clustering, but conceptually, it can be used in other combinatorial optimization problems as well. Initial results in k-way graph/hypergraph partitioning [32], [33] look good. We are in the process of investigating its performance in such application areas as well.

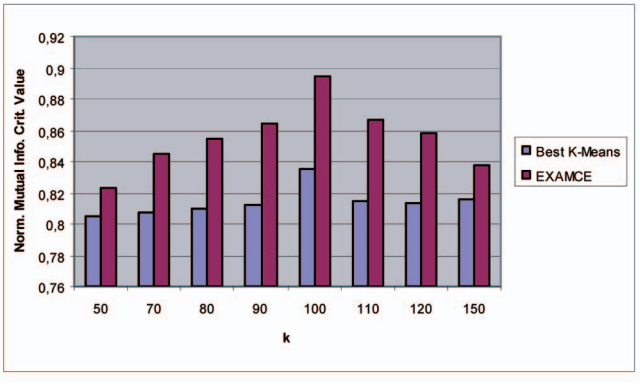


Fig. 8. Clustering accuracy of K-means versus EXAMCE on "Gauss100" for various k sought.

References

[1] P. Hansen and N. Mladenovic, "J-Means: A New Local Search Heuristic for Minimum Sum-of-Squares Clustering," *Pattern Recognition*, vol. 34, no. 2, pp. 405-413, Feb. 2001.

[2] M. Laszlo and S. Mukherjee, "A Genetic Algorithm Using Hyper-Quadtrees for Low-Dimensional K-Means Clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 533-543, Apr. 2006.

[3] J. Pacheco, "A Scatter-Search Approach for the Minimum-Sum-of-Squares Clustering Problem," *Computers and Operations Research*, vol. 32, no. 5, pp. 1325-1335, May 2005.

[4] T. Lange and J.M. Buhmann, "Combining Partitions by Probabilistic Label Aggregation," *Proc. Int'l Conf. Knowledge Discovery in Databases*, 2005.

[5] A.K. Jain and A. Fred, "Evidence Accumulation Clustering Based on the K-Means Algorithm," *Structural, Syntactic, and Statistical Pattern Recognition*, pp. 442-451, Springer, 2002.

[6] B. Fischer and J.M. Buhmann, "Bagging for Path-Based Clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 11, pp. 1411-1415, Nov. 2003.

[7] H. Ayad and M. Kamel, "Cumulative Voting Consensus Method for Partitions with Variable Number of Clusters," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 1, pp. 160-173, Jan. 2008.

[8] E. Dimitriadou, A. Weingessel, and K. Hornik, "A Combination Scheme for Fuzzy Clustering," *Int'l J. Pattern Recognition and Artificial Intelligence*, vol. 16, no. 7, pp. 901-912, 2002.

[9] L. Breiman, "Bagging Predictors," Technical Report 421, Dept. of Statistics, Univ. of California at Berkeley, 1994.
[10] L.I. Kuncheva, *Combining Pattern Classifiers.* Wiley, 2004.
[11] A. Strehl and J. Ghosh, "Cluster Ensembles—A Knowledge Re-Use Framework for Combining Multiple Partitions," *J. Machine Learning Research,* vol. 3, pp. 583-618, 2002.
[12] A. Topchy, A.K. Jain, and W. Punch, "Clustering Ensembles: Models of Consensus and Weak Partitions," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 12, pp. 1866-1881, Dec. 2005.
[13] A. Asuncion and D.J. Newman, "UCI Machine Learning Repository," School of Information and Computer Science, Univ. of California, http://www.ics.uci.edu/~mlearn/MLRepository.html, 2007.
[14] L.I. Kuncheva and D.P. Vetrov, "Evaluation of Stability of K-Means Cluster Ensembles with Respect to Random Initializations," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 11, pp. 1798-1808, Nov. 2006.
[15] M. Meila, "Comparing Clusterings: An Axiomatic View," *Proc. 22nd Int'l Conf. Machine Learning,* pp. 577-584, 2005.
[16] V. Singh, L. Mukherjee, J. Peng, and J. Xu, "Ensemble Clustering Using Semidefinite Programming," *Advances in Neural Information Processing Systems,* J.C. Platt, D. Koller, Y. Singer, and S. Roweis, eds., pp. 1353-1360, MIT Press, 2008.
[17] S. Theodoridis and K. Koutroumbas, *Pattern Recognition,* third ed. Academic Press, 2006.
[18] H. Li, K. Zhang, and T. Jiang, "Minimum Entropy Clustering and Applications to Gene Expression Analysis," *Proc. IEEE Conf. Computational Systems Bioinformatics,* pp. 142-151, 2004.
[19] P. Hansen and N. Mladenovic, "Variable Neighborhood Search for the P-Median," *Location Science,* vol. 5, no. 4, pp. 207-226, 1997.
[20] M.G.C. Resende and R.F. Werneck, "A Hybrid Heuristic for the P-Median Problem," *J. Heuristics,* vol. 10, pp. 59-88, 2004.
[21] D. Pelleg and A. Moore, "X-Means: Extending K-Means with Efficient Estimation of the Number of Clusters," *Proc. 17th Int'l Conf. Machine Learning,* pp. 727-734, 2000.
[22] O. du Merle, P. Hansen, B. Jaumard, and N. Mladenovich, "An Interior Point Algorithm for Minimum Sum of Squares Clustering," *SIAM J. Scientific Computing,* vol. 21, no. 4, pp. 1484-1505, Mar. 2000.
[23] P. Hansen and B. Jaumard, "Cluster Analysis and Mathematical Programming," *Math. Programming,* vol. 79, pp. 191-215, 1997.
[24] G.L. Nemhauser and L.A. Wolsey, *Integer and Combinatorial Optimization,* first ed. Wiley Interscience, 1988.
[25] G.B. Dantzig and P. Wolfe, "Decomposition Principle for Linear Programs," *Operations Research,* vol. 8, no. 1, pp. 101-111, 1960.
[26] P.S. Bradley, K.P. Bennett, and A. Demiriz, "Constrained K-Means Clustering," Microsoft Research Technical Report MSR-TR-2000-65, May 2000.
[27] T. Achterberg, "Constraint Integer Programming," PhD thesis, Technische Univ. Berlin, 2007.
[28] J. Peng and Y. Xia, "A New Theoretical Framework for K-Means-Type Clustering," *Foundations and Advances in Data Mining,* W. Chu and T.Y. Lin, eds., pp. 79-95, Springer, 2005.
[29] K. Rose, "Deterministic Annealing for Clustering, Compression, Classification, Regression and Related Optimization Problems," *Proc. IEEE,* vol. 86, no. 11, pp. 2210-2239, Aug. 1998.
[30] B. Akteke-Ozturk, G.-W. Weber, and E. Kropat, "Continuous Optimization Approaches for Clustering via Minimum Sum of Squares," *Proc. 20th Mini-EURO Conf. Continuous Optimization and Knowledge-Based Technologies,* May 2007.
[31] E.B. Baum, "Toward Practical 'Neural' Computation for Combinatorial Optimization Problems," *Neural Networks for Computing,* J. Denker, ed., Am. Inst. of Physics, 1986.
[32] G. Karypis and V. Kumar, "Multi-Level k-Way Hyper-Graph Partitioning," *VLSI Design,* vol. 11, no. 3, pp. 285-300, 2000.
[33] I.T. Christou and R.R. Meyer, "Decomposition Algorithms for Communication Minimization in Parallel Computing," *Non-Linear Assignment Problems Theory and Practice,* P.M. Pardalos and L. Pitsoulis, eds., Kluwer Academic Publishers, 2000.

Ioannis T. Christou received the DipIIng degree in electrical engineering from the National Technical University of Athens, Greece, in 1991, the MSc and PhD degrees in computer sciences from the University of Wisconsin at Madison in 1993 and 1996, respectively, and the MBA degree from NTUA and Athens University of Economics and Business in 2006. He has been with Delta Technology, Inc., as a senior developer, with Intracom S.A. as an area leader in data and knowledge rngineering, and with Lucent Technologies Bell Labs as a member of the technical staff. He has also been an adjunct assistant professor with the Department of Computer Engineering and Informatics, University of Patras, Greece. He is currently an associate professor at Athens Information Technology and an adjunct professor at Carnegie-Mellon University, and has published many articles in scientific journals and peer-reviewed conferences. He is a member of the ACM and the Technical Chamber of Greece. He is a member of the IEEE and the IEEE Computer Society.