Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering
http://pig.sagepub.com/

On the detection of nearly optimal solutions in the context of single-objective space mission design problems

O Schütze, A Lara, C A Coello Coello and M Vasile

Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering 2011 225: 1229
originally published online 23 September 2011
DOI: 10.1177/0954410011413693

On the detection of nearly optimal solutions in the context of single-objective space mission design problems

O Schütze[1], **A Lara**[1], **C A Coello Coello**[1], and **M Vasile**[2*]
[1]Computer Science Department, CINVESTAV-IPN, Mexico City, Mexico
[2]Department of Mechanical Engineering, University of Strathclyde, Glasgow, UK

The manuscript was received on 2 November 2010 and was accepted after revision for publication on 24 May 2011.

DOI: 10.1177/0954410011413693

Abstract: When making decisions, having multiple options available for a possible realization of the same project can be advantageous. One way to increase the number of interesting choices is to consider, in addition to the optimal solution x^*, also nearly optimal or approximate solutions; these alternative solutions differ from x^* and can be in different regions – in the design space – but fulfil certain proximity to its function value $f(x^*)$. The scope of this article is the efficient computation and discretization of the set E of ϵ–approximate solutions for scalar optimization problems. To accomplish this task, two strategies to archive and update the data of the search procedure will be suggested and investigated. To make emphasis on data storage efficiency, a way to manage significant and insignificant parameters is also presented. Further on, differential evolution will be used together with the new archivers for the computation of E. Finally, the behaviour of the archiver, as well as the efficiency of the resulting search procedure, will be demonstrated on some academic functions as well as on three models related to space mission design.

Keywords: single objective optimization, approximate solutions, differential evolution, space mission design

1 INTRODUCTION

One common way to solve a real world engineering problem is by transforming it into an optimization problem (without loss of generality, we assume in the sequel minimization problems) and to seek for the (or at least one) optimal solution. From a practical point of view, however, it can, in some cases, make sense to include (in addition to the optimal solutions) also nearly optimal solutions since this will give the decision maker (DM) a larger variety of possibilities: two solutions which are 'near' in objective space (i.e. have similar objective values) may differ significantly in parameter space. The storage of both solutions may give the DM a second option for the realization of his/her project.

As one example, consider the objective shown in Fig. 1. In case, the DM is willing to accept a deterioration of $\epsilon \in \mathbb{R}_+$, the objective function f contains next to the global minimizer x_1 also the local minimizer x_2 which is such an 'ϵ-approximate solution' (i.e. the function values of $f(x_1)$ and $f(x_2)$ differ by less than ϵ). All other points in $[a, b] \cup [c, d]$ are also approximate solutions, however, they are all 'dominated' (i.e. they offer worse values of f) within their connected components by the solutions x_1 and x_2. Hence, these solutions are possibly too near to

*Corresponding author: Space Advanced Research Team, Department of Mechanical Engineering, University of Strathclyde, 75 Montrose Street, Glasgow G1 1XJ UK.
email: massimiliano.vasile@strath.ac.uk*

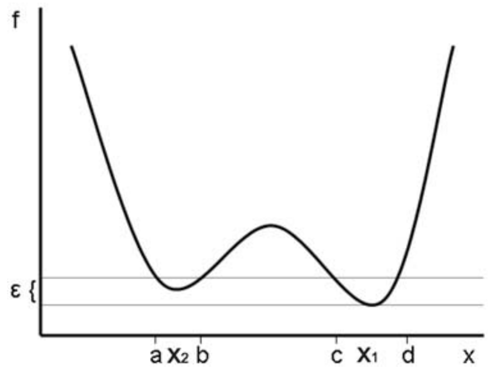


Fig. 1 Example of an objective function with two local minima x_1 and x_2 which are similar in objective space but differ in parameter space

these two points in order to give the DM a significant new alternative. This depends on the minimal distance that these two solutions must keep from each other in order to be considered as 'distinct' from a practical point of view.

Hence, an 'optimal' outcome of the optimization process (depending on the problem) could be to present the possible choices x_1 and x_2 – and no other solution to avoid confusing the DM and for the sake of an efficient computation (since no superfluous options have to be stored and updated).

As another example, consider the problem of designing an 'optimal' trajectory from Earth to the comet 67P/Churyumov-Gerasimenko (see references [**1**, **2**], and also section 5.4). One crucial parameter is the launch date T_0 which is in the time window [1460, 1825] MJD2000 (Modified Julian Date 2000). The best known solution is a trajectory P_1 with $T_0(P_1) = 1546$ [MJD2000] (value rounded) and objective value $f(P_1) \approx 1.34$ [km/s] (measured as the total variation in velocity that the engines have to deliver to reach the destination). If the DM is willing to accept a deterioration of $\epsilon = 0.5$ [km/s], then he/she is given (among others) another two possible local optimal trajectories P_2 (with $T_0(P_2) = 1619$ [MJD2000] and $f(P_2) = 1.76$ [km/s]) and P_3 (with $T_0(x_3) = 1748$ [MJD2000] and $f(x_3) = 1.76$ [km/s]). Hence, in that case, the DM is offered two more choices for the launch of the spacecraft (2.5 respectively 6.5 months after $T_0(P_1)$).

In this article, the problem of computing approximate solutions of scalar optimization problems is addressed. Since the set E of these ϵ-approximate solutions typically forms a n-dimensional set, where n is the dimension of the parameter space, a suitable discretization is mandatory in order to be applicable to real world problems. In this study, the focus will be on the approximation of the local minima within E. However, also further points will be considered. If, for instance, the objective is 'flat' around a local minimum in E (as, for instance, happens for the 'funnels' in models related to space mission design), then also points which are not locally optimal but differ sufficiently in parameter space from the local solutions could be interesting for the DM. To achieve this goal, two archiving strategies (i.e. strategies to maintain a subset of the obtained data) will be proposed and investigated. In order to obtain an efficient algorithm for the approximation of E, the archivers will be combined with differential evolution (DE), a heuristic that has already shown its efficiency on space mission design problems [**3**, **4**].

This study can be considered as an 'extension' of previous studies on the computation of approximate solutions for multi-objective optimization problems (MOPs) [**5–7**]. The crucial difference when considering scalar optimization problems (i.e. one objective) is that in that case, a discretization in parameter space can be performed. As will be seen later on, a discretization of the set of interest is mandatory. In case multiple objectives are under consideration, a discretization in parameter space leads either to a tremendous number of archive entries when choosing small or even moderate values for the discretization parameter, or leads to grave loss of information in case this parameter is large. The latter is due to the fact that the solution set (the so-called *Pareto set*) typically forms a $(k-1)$-dimensional object, where k is the number of objectives in the MOP, and hence, a discretization around a promising point (optimal or nearly optimal) leads to a non-observance of an entire (and large) optimal region. This will change, however, if we consider only one objective since in that case the (local or global) optima are typically isolated (as in Fig. 1). Thus, in such case, a discretization can in principle be performed in parameter space without essential loss of information. A preliminary study of this can be found in reference [**8**].

Next, there is a certain relation to multi-modal optimization [**9–17**], where the task is to detect all local minima within a given region. However, note that there are some differences to the approach in this study: first, this study is not interested in local minima nor any other point outside E. Second, and that is more important, the present study is not 'restricted' to local minima (though better solutions in a given neighbourhood will be preferred in order to discretize the set of interest E). For this, consider for instance Rosenbrock's banana function (which indeed shares some characteristics with the objectives related to space mission design considered in this study). The function contains one global minimum m which is located inside a long, narrow, and flat valley. Hence, it could make sense to compute next to m (as for multi-modal optimization) also further approximate solutions along the valley, since they could be distinct solutions for the DM.

Finally, approximate solutions in space mission design problems have already been considered in reference [**18**], where a hybrid multiagent approach has been chosen for their detection.

The remainder of this article is organized as follows: section 2 gives the required background for the understanding of the sequel. Section 3 presents and investigates the set of interest, and in section 4 methods are proposed for their efficient computation. Section 5 presents some numerical results, and finally, some conclusions are drawn in section 6.

2 BACKGROUND

In this article, we consider single-objective optimization problems (SOPs) of the following form:

$$\min_{x \in Q} f(x) \tag{1}$$

where it is assumed that $f: Q \subset \mathbb{R}^n \rightarrow \mathbb{R}$ is continuous. For theoretical purposes, it has to be assumed that f is even continuously differentiable, though this smoothness assumption will never be used in the numerical treatment (since DE does not exploit gradient information). Further, it has to be assumed that the domain Q is compact which is typically given if Q is defined by equality and inequality constraints. In the easiest case (which is already sufficient for the models considered in this study), Q can be defined by box-constraints, i.e. the domain forms an n-dimensional box

$$Q = \left\{ x \in \mathbb{R}^n \;:\; a_i \leqslant x_i \leqslant b_i,\ i = 1, \ldots, n \right\} \tag{2}$$

where a_i and b_i are the lower and upper bounds of each parameter x_i.

The solution set of (1) is given by

$$M_Q := \{x \in Q \;:\; f(x) \leqslant f(y)\ \forall y \in Q\} \tag{3}$$

Note that M_Q does not have to consist of one single solution, however, except for plateau functions, the solution set will be a finite set of points (i.e. a 0-dimensional set). To illustrate this, the reader may think of the sine curve restricted to a closed interval.

Algorithm 1 gives a framework of a generic stochastic search algorithm, which has first been studied in reference [**19**], and which will be considered in this study. Here, $Q \subset \mathbb{R}^n$ denotes the domain of the problem, P_j the candidate set (or population) of the generation process at iteration step j, and A_j the corresponding archive.

Algorithm 1 Generic Stochastic Search Algorithm

$P_0 \subset Q$ drawn at random
$A_0 = \mathit{ArchiveUpdate}(P_0, \emptyset)$
for $j = 0, 1, 2, \ldots$ **do**
  $P_{j+1} = \mathit{Generate}(P_j)$
  $A_{j+1} = \mathit{ArchiveUpdate}(P_{j+1}, A_j)$
endfor

Finally, some distances between points and sets as well as between different sets are defined which will be needed to evaluate the approximation quality of the outcome set.

Definition 2.1

Let $u, v \in \mathbb{R}^n$ and $A, B \subset \mathbb{R}^n$. The semi-distance $\mathrm{dist}(\cdot,\cdot)$ and the Hausdorff distance $d_H(\cdot,\cdot)$ are defined as follows:

(a) $\mathit{dist}(u, A) := \inf_{v \in A} \| u - v \|$;
(b) $\mathit{dist}(B, A) := \sup_{u \in B} \mathit{dist}(u, A)$;
(c) $d_H(A, B) := \max \{\mathit{dist}(A, B), \mathit{dist}(B, A)\}$.

3 THE SET OF INTEREST

In the following, the set of interest, $M_{Q,\epsilon}$, is defined and some of its topological properties are discussed.

Definition 3.1

Let $\epsilon > 0$. $x \in Q$ is called an ϵ-approximate solution of (1) if $f(x) - \epsilon \leqslant f(y)$ for all $y \in Q$. The set of ϵ-efficient solutions $M_{Q,\epsilon}$ of (1) is defined by

$$M_{Q,\epsilon} = \{x \in Q \;:\; f(x) - \epsilon \leqslant f(y)\ \forall y \in Q\} \tag{4}$$

A point x is an ϵ-approximate solution of a set A if $f(x) - \epsilon \leqslant f(a)$ for all $a \in A$.

The following examples illustrate the set of interest.

Example 3.2

(a) Let $f: \mathbb{R}^n \rightarrow \mathbb{R}$ be given by

$$f(x) = \sum_{i=1}^{n} x_i^2 \tag{5}$$

then the sets M_Q and $M_{Q,\epsilon}$ for an $\epsilon > 0$ are given by

$$M_Q = \{0\}, \quad M_{Q,\epsilon} = \left\{ x \in \mathbb{R}^n \;:\; \sum_{i=1}^{n} x_i^2 \leqslant \epsilon \right\} \tag{6}$$

that is, $M_{Q,\epsilon}$ is the closed ball with centre 0 and radius $\sqrt{\epsilon}$.

(b) The set of ϵ-approximate solutions for the introductory example (Fig. 1) is given by $M_{Q,\epsilon} = [a, b] \cup [c, d]$, i.e. in particular disconnected.

The following short discussion shows that $M_{Q,\epsilon}$ is typically n-dimensional (whereas M_Q is typically 0-dimensional): let $x^* \in M_Q \cap \mathring{Q}$, where $\mathring{Q}$ denotes the interior of Q, and f is continuous. Then, there exists, by continuity of f, a neighbourhood N of x^* inside Q such that

$$f(x) - \epsilon \leqslant f(x^*) \quad \forall x \in N \tag{7}$$

and hence, the n-dimensional set N is contained in $M_{Q,\epsilon}$. Thus, suitable discretization strategies are required for the efficient use of approximate solutions.

Another important aspect is the connectedness of the set of interest. It can be shown (analogue to reference [**20**]) that in case both the objective f as well as the domain Q are convex, then $M_{Q,\epsilon}$ is connected (and can possibly be computed most efficiently by local search procedures if at least one solution is available), but this does not hold in general, as the above example shows. Hence, global strategies seem to be advantageous for the treatment of general objectives.

One potential problem at least for theoretical observations is that $M_{Q,\epsilon}$ may contain isolated points. For this, consider the objective function shown in Fig. 2 which is a modification of the introductory example: in this case, it is $M_{Q,\epsilon} = \{x^*\} \cup [c, d]$, that is, contains the isolated point x^*. The problem with such points is that it cannot be guaranteed to capture them by the use of stochastic search algorithms [**20**]. To allow convergence of the algorithm the following has to be assumed

$$B \subset \mathring{Q} \quad \text{and} \quad \nabla f(x) \neq 0 \; \forall x \in B \tag{8}$$

where $\nabla f(x)$ denotes the gradient of f at x, and B the boundary of $M_{Q,\epsilon}$, i.e. it holds

$$B := \{x \in Q \mid f(m) + \epsilon = f(x) \text{ for } m \in M_Q\} \tag{9}$$

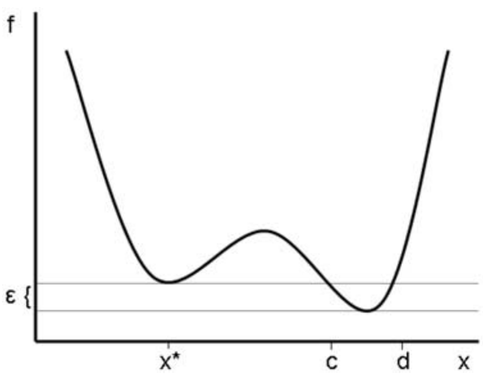


Fig. 2 Example where the set of interest contains an isolated point: here, it is $M_{Q,\epsilon} = \{x^*\} \cup [c, d]$

Under this assumption, it can be shown (analogue to reference [**20**] that $M_{Q,\epsilon}$ contains no isolated points, that is

$$\overline{\mathring{M}_{Q,\epsilon}} = \overline{M_{Q,\epsilon}} \tag{10}$$

Finally, it is important to note that the approach can be used to detect multiple solutions in M_Q since every optimal solution is also an ϵ-approximate solution. To be more precise, the set of optima M_Q is contained in $M_{Q,\epsilon}$ for every $\epsilon > 0$. Furthermore, it is

$$M_Q = \bigcap_{\epsilon > 0} M_{Q,\epsilon} \tag{11}$$

Classical elitist approaches have strong limitations in detecting multiple solutions since there is typically only one 'best' (scalar) value out of a finite set of candidates. Regarding this, it is important to note that a discretization of $M_{Q,\epsilon}$ cannot be performed by merely considering the objective values (as e.g. done in reference [**20**] for the multi-objective case).

4 AN ALGORITHM FOR THE APPROXIMATION OF $M_{Q,\epsilon}$

In this section, one possibility to compute approximations of $M_{Q,\epsilon}$ – DE together with an external archive – is presented. Following the notation of Algorithm 1, the archiver and the generator which constitute the stochastic search process will be considered separately.

4.1 Two archiving strategies

In the following, two possible archiving strategies aiming for the representation of $M_{Q,\epsilon}$ are discussed: the first captures all ϵ-approximate solutions out of the obtained data, and the second one uses a certain discretization strategy.

The first archiver considered here, *ArchiveUpdate*$M_{Q,\epsilon}$, is shown in Algorithm 2. The information management is straightforward: the algorithm captures all the ϵ-efficient solutions out of the obtained data (i.e. out of the sequence of candidate sets P_i). The following proposition states this more precisely.

Proposition 4.1

Let $l \in \mathbb{N}$, $\epsilon \in \mathbb{R}_+$, $P_1, \ldots, P_l \subset \mathbb{R}^n$ be finite sets, and A_i, $i = 1, \ldots, l$, be obtained by *ArchiveUpdate*$M_{Q,\epsilon}$ as in Algorithm 1. Then

$$A_l = M_{C_l,\epsilon} = \{x \in C_l \; : \; f(x) - \epsilon \leqslant f(y) \; \forall y \in C_l\} \tag{12}$$

where $C_l = \bigcup_{i=1}^{l} P_i$.

Proof

Follows by construction of *ArchiveUpdateM*$_{Q,\epsilon}$. ■

Next, the limit behaviour of the sequence of archives A_i generated by the archiver is investigated. For this, the following assumption on the generation process has to be made [**21**, **22**]

$$\forall x \in Q \text{ and } \forall \delta > 0 : \quad P(\exists l \in \mathbb{N} \; : \; P_l \cap B_\delta(x) \cap Q \neq \emptyset) = 1 \tag{13}$$

where $P(\mathring{A})$ denotes the probability for event A. Assumption (13) says, roughly speaking, that every neighbourhood $U \cap Q$ of every point gets 'visited' by *Generate*() after finitely many steps with probability one. The following consideration shows that we cannot assume less: if (13) does not hold, there exists with probability one a point $x \in Q$ and a neighbourhood $\tilde{U} = U \cap Q$ of x such that no candidate solution $p \in P_l$ lies in $\tilde{U}$ for all $l \in \mathbb{N}$. Thus, no convergence can be guaranteed since a part of $M_{Q,\epsilon}$ can be contained in $\tilde{U}$ which is never 'visited'.

Corollary 4.2

Let a SOP $f: \mathbb{R}^n \to \mathbb{R}$ be given, where F is continuous, let $Q \subset \mathbb{R}^n$ be a compact set and $\epsilon \in \mathbb{R}_+$. Further, let the assumptions (8) and (13) be fulfilled. Then, an application of Algorithm 1, where *ArchiveUpdateM*$_{Q,\epsilon}$ is used to update the archive, leads to a sequence of archives A_l, $l \in \mathbb{N}$, with

$$\lim_{l \to \infty} d_H(M_{Q,\epsilon}, A_l) = 0, \quad \text{with probability one} \tag{14}$$

Proof

The proof is analogue to the proof of Theorem 2 of reference [**5**] using the modified assumption (8). ■

However, due to the dimension of $M_{Q,\epsilon}$, the strategy is apart from a theoretical point of view only interesting, e.g. if the cost of a function evaluation is relatively high, i.e. if only a moderate amount of function calls can be spent within a given time budget. In that case, it makes sense to store *all* interesting information (and not to lose single promising candidates due to discretization) and *ArchiveUpdateM*$_{Q,\epsilon}$ can be chosen without significant computational loss.

More interesting – and mandatory for the efficient application to real world problems – is certainly to filter the incoming data farther by considering a suitable discretization strategy. In order to accomplish this task, *ArchiveUpdateM*$_{Q,\epsilon}D_x$ (Algorithm 3) is proposed here which is similar to the first archiver but performs a selection of the promising data. The underlying idea of *ArchiveUpdateM*$_{Q,\epsilon}D_x$ is to keep (locally) best found solutions within a certain range (using $\epsilon \in \mathbb{R}_+$ in objective space and a vector $\Delta \in \mathbb{R}^n_+$ in parameter space) and to discard inferior points in the neighbourhood of these ones in order to obtain a suitable discretization (compare to the motivating example in section 1).

Algorithm 2 $A := $ *ArchiveUpdateM*$_{Q,\epsilon}$ (A_0, P, ϵ)

Require: archive A_0, candidate set $P \subset Q$, tolerance $\epsilon \in \mathbb{R}_+$
Ensure: updated archive A

```
A := A_0
for all p ∈ P do
  if ∄a ∈ A : f(a) + ε ≤ f(p) then
    A := A ∪ {p}
  end if
  for all a ∈ A do
    if f(p) + ε < f(a) then
      A := A\{a}
    end if
  end for
end for
```

More precisely, given an archive A_0 and a candidate solution p, the new archiver A is constructed as follows: p is rejected (and hence, A is set to A_0) if either p is not an ϵ-approximate solution of A_0 (i.e. $f(x_b) + \epsilon < f(p)$, where x_b is the best found solution), or if there exists an element $a \in A_0 \cap B^\infty_\Delta(p)$, where the neighbourhood $B^\infty_\Delta(p)$ is defined as

$$B^\infty_\Delta(p) := \{x \in \mathbb{R}^n \; : \; |x_i - p_i| < \Delta_i, \; i = 1, \ldots, n\} \tag{15}$$

which is at least as good as p (line 6 of Algorithm 4). If p is not discarded, this means that (i) this point is an ϵ-approximate solution of A, and (ii) that it is the best point in its neighbourhood (the latter defined by $\Delta \in \mathbb{R}^n_+$). Hence, the new archive A consists of p as well as all other points of A_0 which are ϵ-approximate solutions of p, and which are not in the Δ-neighbourhood of p (lines 10-14 of Algorithm 3).

Note that *ArchiveUpdateM*$_{Q,\epsilon}D_x$ in Algorithm 3 is formulated for the consideration of one candidate point p. However, an extension to entire sets $P \subset Q$ is straightforward. Further, for the sake of a better readability the best found solution x_b is explicitly stated. This is, in fact, not required since the best found solution is always included in the archive due to the construction of Algorithm 3.

Algorithm 3 $\{A, x_b\} := ArchiveUpdateM_{Q,\epsilon}D_x\ (A_0, x_{b,0}, p, \epsilon, \Delta)$

Require: archive A_0, best found solution $x_{b,0}$, candidate solution $p \in Q$, tolerance $\epsilon \in \mathbb{R}_+$, discretization parameter $\Delta \in \mathbb{R}^n_+$
Ensure: updated archive A, best found solution x_b
1: **if** $f(p) < f(x_{b,0})$ **then**
2: $x_b := p$
3: **else**
4: $x_b := x_{b,0}$
5: **end if**
6: **if** $f(x_b) + \epsilon < f(p)$ or $(\exists a \in A_0 : p \in B^{\infty}_{\Delta}(a_j)$ and $f(a) \leqslant f(p))$ **then**
7: $A := A_o$ $\lhd$ discard p
8: **return**
9: **end if**
10: $A := \{p\}$
11: **for all** $a \in A_0$ **do**
12: **if** $f(a) \leqslant f(x_b) + \epsilon$ and $a \notin B^{\infty}_{\Delta}(p)$ **then**
13: $A := A \cup \{a\}$
14: **end if**
15: **end for**

Results of the sequence of archives when using $ArchiveUpdateM_{Q,\epsilon}D_x$ are not as straightforward as for the first archiver $ArchiveUpdateM_{Q,\epsilon}$. Given A_l and C_l as above, and denoting by $x_{b,l}$ to the best found solution in step l, then it holds

$$x_{b,l} \in M_{C_l} \quad \text{and} \quad A_l \subset M_{C_l,\epsilon} \tag{16}$$

However, further approximation qualities (such as the Hausdorff distance between A_l and $M_{C_l,\epsilon}$) for finite candidate solutions $\{p_1, \ldots, p_s\}$, $s \in \mathbb{N}$, cannot be given since the final archive A_l depends on the order the candidate solutions p_i are considered. For this, consider the following example: let $f : [0, 10] \to \mathbb{R}$, $f(x) = x^2$, and $\epsilon = 1$. Then, it is $M_{Q,\epsilon} = [0, 1]$. Let a hypothetical candidate set be given by

$$P_1 = \{0, 0.05, 0.1, \ldots, 0.95, 1\} \tag{17}$$

and $\Delta = 0.1$. Then, an application of $ArchiveUpdateM_{Q,\epsilon}D_x$, where $A_0 := \emptyset$ and the entries of P_1 are considered in ascending order, leads to the archive

$$A^{(1)}_{21} = \{0, 0.1, 0.2, \ldots, 1\} \tag{18}$$

If the entries of P_1 are considered instead in descending order, then the final archive is given by

$$A^{(2)}_{21} = \{0\} \tag{19}$$

since in each iteration the actual candidate point is added to the archive while the previous one is deleted. However, since by assumption on the generator, each region is (re-)visited after finitely many steps, the 'limit archive' (i.e. for iteration step $l \to \infty$) in this case will be equal to $A^{(1)}_{21}$.

The following result shows that local minima within $M_{Q,\epsilon}$ will be approximated under certain assumptions (and also explain the 'limit archive' of the above example):

Proposition 4.3

Let $m \in \mathring{M}_{Q,\epsilon}$ be the unique minimum of f within the domain $Q \cap B^{\infty}_{\tilde{\Delta}}(m)$, where $\tilde{\Delta}_i > \Delta_i$, $i = 1, \ldots, n$. Then an application of Algorithm 1, where $ArchiveUpdateM_{Q,\epsilon}D_x$ is used to update the archive, leads to a sequence of archives A_l, $l \in \mathbb{N}$ such that with probability one

(a) $\exists a_l \in A_l : a_l \to m$ *for* $l \to \infty$
(b) $A_l \cap B^{\infty}_{\Delta}(m) = \{a_l\} \quad \forall l \geqslant l_0$ *for an integer* l_0

Proof

Ad (a): By assumption (13) on the generator there exists with probability one a sequence of candidate solutions $p_{l_i} \in Q$ such that $p_{l_i} \in M_{Q,\epsilon}$ and $p_{l_i} \to m$ for $i \to \infty$. By construction of $ArchiveUpdateM_{Q,\epsilon}D_x$, the candidate solution p_{l_i} is either discarded if there already exists an archive entry $a_{l_i} \in A_{l_i}$ with $f(a_{l_i}) \leqslant f(p_{l_i})$ (line 6 of Algorithm 3), or $a_{l_i} := p_{l_i}$ is added to the archive (line 10 of Algorithm 3). Since entries $a \in A_l$ are only replaced from the archive if there is a better solution in the Δ-neighbourhood of a, there exists with probability one for *all* $l \in \mathbb{N}$ an archive entry $a_l \in A_l \cap B^{\infty}_{\Delta}(m))$ such that $f(a_l) \to f(m)$ for $l \to \infty$. Since m is the unique solution in $Q \cap B^{\infty}_{\tilde{\Delta}}(m)$, it follows that $a_l \to m$ for $l \to \infty$ with probability one.
Ad (b): follows by (a) and the construction of $ArchiveUpdateM_{Q,\epsilon}D_x$. ■

Crucial for the successful application of the latter archiver is certainly the proper choice of Δ. By construction of the archiver, it holds for every archive entry $a \in A_l$

$$A \cap B^{\infty}_{\Delta}(a) = \{a\} \tag{20}$$

and hence, the choice of Δ has a direct influence on the distribution of the archive entries (see e.g. the numerical results in section 5.2). In general, smaller values lead to a better approximation quality (measured in the Hausdorff sense). However, too small values should be avoided in order to prevent huge archive sizes: assume, for simplicity, that $\Delta = (\delta, \ldots, \delta)$, where $\delta \in \mathbb{R}_+$ is 'small'. Then, we expect due to the dimension of $M_{Q,\epsilon}$ that the magnitude of the limit archive is of the order $O(\delta^{-n})$. Larger entries of Δ lead to the focus – and in the ideal case also to a complete reduction – of the local minima within $M_{Q,\epsilon}$ (i.e. $O(1)$ entries in the limit archive), however, the possibility increases that several minima are located within one Δ-neighbourhood.

In case the objective f is derived from a real world problem, a possible rule of thumb is to choose the entries of Δ such that two solutions x_1 and x_2 within the same set $B_{\Delta}^{\infty}(x)$ do not represent different options for the DM. As an example, consider the departure time T_0 of a trajectory design problem. If two trajectories are given where the departure time does not differ significantly (say, less than 1 week), the two trajectories cannot be regarded as different (at least according to T_0), and the choice would always be in favour of the best of both trajectories (i.e. the inferior trajectory does not have to be stored). In this manner, the required number of archive entries depends on the behaviour of f and the preferences of the DM.

4.2. Using differential evolution as our generator

Having stated the archiver, it remains to define the generator in order to obtain a complete search procedure as defined in Algorithm 1. The most important aspect for the generation process – next to convergence to (local) minima – is a good exploratory behavior. Hence, a population-based method seems to be most promising. Here, we have chosen to utilize DE as the basis for the generator procedure. This state of the art heuristic has shown its efficiency on a variety of scalar optimization problems – including problems related to space mission design [**3**].

Algorithm 4 shows the complete search procedure. As it can be seen, the outcome of DE is simply used to feed the archiver with candidate solutions. We have observed that this already defines a satisfying search engine. In the implementations, we have used the classical variant of DE as described in reference [**23**].

For future investigation, there are several issues to be explored. For instance, it seems promising to couple the population of DE with entries of the archive. Our studies have shown, however, that this is a non-trivial task for practical use of the algorithm due to a performance decrease in case the archive contains many elements. Further, we do not exclude here the possibility that other DE variants or even other population-based methods can yield similar or even better results. Note, however, that the focus of this work is on the choice of the archiver. The subsequent results show that the chosen approach is suitable enough for the treatment of the problems that interest us.

Algorithm 4 DE + $\mathit{ArchiveUpdateM}_{Q,\epsilon}D_x$

```
procedure DE
  A_0 = ArchiveUpdate(P_0, ∅).
  Generate a random initial population P_0.
  for j = 0, 1, 2, ..., do
    Apply the DE operators to P_j in order to get
    a new population P_{j+1}.
    for every p ∈ P_{j+1} do
      A_j := ArchiveUpdate(p, A_j).
    end for
    A_{j+1} := A_j
  end for
end procedure
```

5 NUMERICAL RESULTS

In the following, some numerical results on two academic problems as well as on three space mission design problems are presented in order to demonstrate the benefit of both the new archiver and the new strategy for the approximation of $M_{Q,\epsilon}$.

5.1 Example A

The first academic function considered here is $f: Q\subset \mathbb{R}^2 \to \mathbb{R}$, where

$$f(x) = \begin{cases} -\sin(x_1)\sin(x_2) & \text{if } (x_1, x_2) \in [0, 10]^2 \\ -\sin(x_1)\sin(x_2) + 1 & \text{otherwise} \end{cases} \tag{21}$$

and domain $Q = [0,\ 200]^2$. The objective is constructed such that the minima are located within $[0, 10]^2$, i.e. $M_Q = \{x_1^*, x_2^*, x_3^*, x_4^*, x_5^*\}$, where

$$x_1^* = \left(\frac{\pi}{2}, \frac{\pi}{2}\right),\ x_2^* = \left(\frac{\pi}{2}, \frac{5\pi}{2}\right),\ x_3^* = \left(\frac{5\pi}{2}, \frac{\pi}{2}\right),$$
$$x_4^* = \left(\frac{3\pi}{2}, \frac{3\pi}{2}\right),\ x_5^* = \left(\frac{5\pi}{2}, \frac{5\pi}{2}\right) \tag{22}$$

If choosing for instance $\epsilon = 0.3$, the set of approximate solutions $M_{Q,\epsilon}$ consists of five connected components, each of them containing one minimizer x_i^*. Further, for $\Delta = (2, 2)$ an 'optimal' archiver A contains exactly five solutions, each of them approximating one minimizer x_i^* (compare to Fig. 3).

In order to compare the result of the novel approach (i.e. DE + archiver), we have chosen to take a multi-start optimization process (using FMINCON of MATLAB, http://www.mathworks.com.) and a random search procedure, both equipped with the archiver $\mathit{ArchiveUpdateM}_{Q,\epsilon}D_x$. We have not considered multi-modal optimizers in this case, since SOP

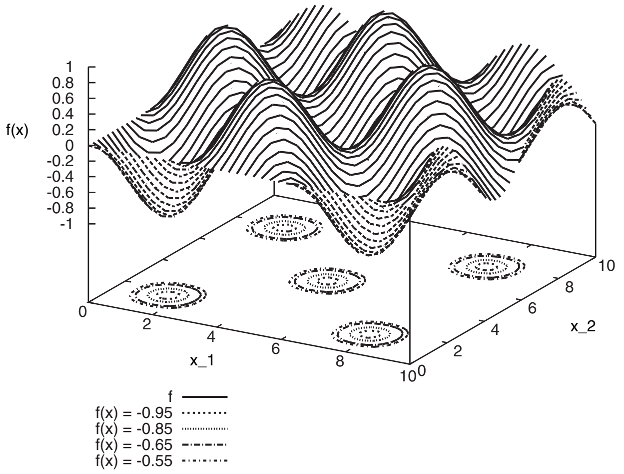


Fig. 3 Surface and contour plot of objective (21) within the ranges $[0, 10]^2$ and the sets $M_{Q,\epsilon}$ for different values of ϵ (the circles around the minimizers x_i^* indicate the boundaries of $M_{Q,\epsilon}$)

Table 1 Number of components found by each method. Minimum, maximum, and average values are over 100 independent runs

Method	Number of components found Min	Mean	Max
Random search	1	2.92	**5**
Multistart (fmincon)	0	1.79	4
Using DE	**4**	**4.97**	**5**

The best values are emphasized in boldface

Table 2 Distance from the archive obtained with each method to the optima set. Minimum, maximum, and average values are over 100 independent runs with at least one component reached. The best values are emphasized in boldface

Method	Dist(A_{final}, M_Q) Min	Mean	Max
Random search	1.54819e-01	7.95134e-01	3.30567e+00
Multistart (fmincon)	**7.18079e-07**	1.48833e-01	3.46062e+00
Using DE	4.17808e-03	**2.96775e-02**	**3.96064e-01**

(21) contains a total of 2000 local minima, but only five of them are contained in $M_{Q,\epsilon}$. Hence, a comparison is not suitable.

Tables 1 to 3 show some averaged numerical results using the three algorithms and a budget of 12 000 function calls per run. For DE, a population size of 200 has been used together with the rand/1 strategy, and the F_{weight} factor of the DE was set to 0.9 in

Table 3 Hausdorff distance between the archive obtained with each method and the optima set. Minimum, maximum, and average values are over 100 independent runs with at least one component reached. The best values are emphasized in boldface

Method	Hausdorff Min	Mean	Max
Random search	6.76238e-01	5.20366e+00	9.32016e+00
Multistart (fmincon)	4.44283e+00	6.28666e+00	1.12152e+01
Using DE	**4.17808e-03**	**2.51149e-01**	**4.44260e+00**

all cases. Table 1 shows the number of connected components detected by each method. Here, DE clearly outperforms the two other methods. This is important to note since the maintainance of diversity is an important issue when considering approximate solutions as motivated in section 1. Tables 2 and 3 are dedicated to the (local) convergence behaviour of the archive entries. Since M_Q consists of five different solutions, the following values have been chosen to be used for a comparison (note that both sets A_{final} and M_Q are finite, and hence, the operators min and max can be used)

$$dist(A_{final}, M_Q) = \max_{a \in A_{final}} \min_{i=1,\ldots,5} \|a - x_i^*\| \tag{23}$$

that is, the maximal distance from an archive entry of A_{final} to M_Q, and the Hausdorff distance

$$d_H(A_{final}, M_Q) = \max(dist(A_{final}, M_Q), dist(M_Q, A_{final})) \tag{24}$$

where

$$dist(M_Q, A_{final}) = \max_{i=1,\ldots,5} \min_{a \in A_{final}} \|x_i^* - a\| \tag{25}$$

Surprisingly, DE can compete with the FMINCON solver when considering $dist(A_{final}, M_Q)$ in this example (and is even better in the mean), and is by far the best when considering the Hausdorff distance. The latter is strongly connected to the result in Table 1.

Since all the local minima of (21) within $M_{Q,\epsilon}$ are also global minima it could be argued that the problem is equal to a 'classical' single-objective optimization problem. To investigate if DE is also able to pull the population toward local optima within $M_{Q,\epsilon}$, which are not global ones we consider the following variation of problem (21)

$$f(x) = \begin{cases} -\sin(x_1)\sin(x_2) - 0.15 & \text{if } \|(x_1, x_2)\|_\infty \leqslant \pi \\ -\sin(x_1)\sin(x_2) & \text{if } \pi < \|(x_1, x_2)\|_\infty \leqslant 10 \\ -\sin(x_1)\sin(x_2) + 1 & \text{otherwise} \end{cases} \tag{26}$$

Table 4 Number of components found by each method (minimum, maximum and average values are over 100 runs, each run with a budget of 12 000 function evaluations) and percentage of runs that reached the component corresponding to the global optima. The best values are emphasized in boldface

Method	Number of components			Percentage of runs reaching the optimal component
	Min	Mean	Max	
Random search	1	2.73	**5**	77
Multi-start (fmincon)	0	1.9	**5**	40
Using DE	**2**	**4.15**	**5**	**100**

For $Q = [0, 200] \times [0, 200]$ and $\epsilon = 0.3$ $M_{Q,\epsilon}$ contains the same five local minima x_1^* to x_5^*, but only x_1^* is a global solution. Table 4 shows a comparison of the components found by each algorithm. The new strategy outperforms the other methods in terms of finding both the global minimum as well as the local minima within $M_{Q,\epsilon}$. Hence, it can be argued that DE is in this case also able to pull the population toward locally optimal solutions.

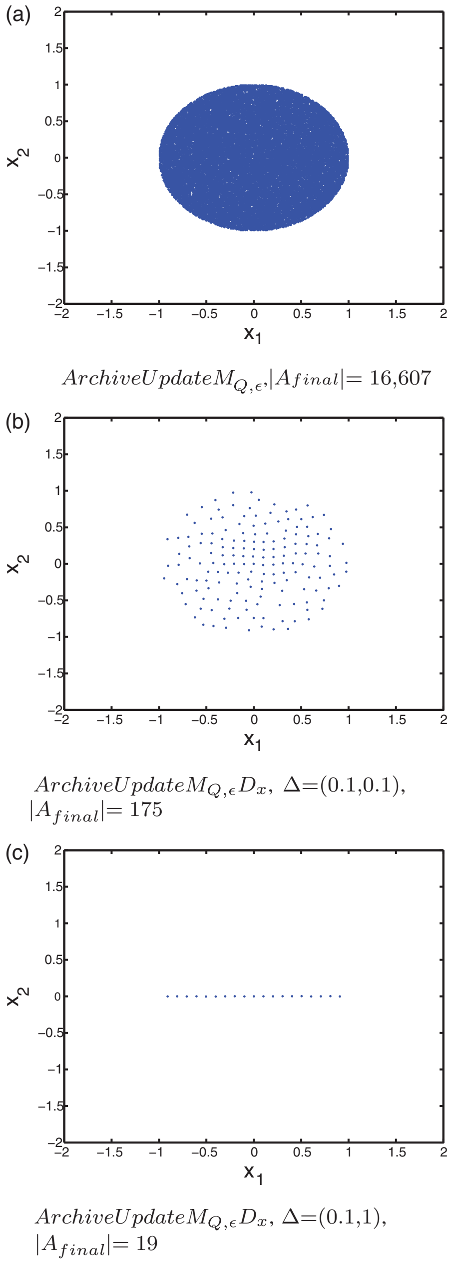
Summarizing, it can be said that the new strategy (DE + *ArchiveUpdate*$M_{Q,\epsilon}D_x$) is efficient in approximating all the local minima of $M_{Q,\epsilon}$ (and only them in this case). However, it has to be noted that the result (i.e. the set of entries which are kept in the archive) highly depends on the choice of ϵ and Δ which is *ad hoc* unclear for this (and in principle for any other) academic model.

5.2 Example B

The next academic function under consideration is (compare to Example 3.2)

$$f : \mathbb{R}^2 \to \mathbb{R}$$
$$f(x) = x_1^2 + x_2^2 \tag{27}$$

Figure 4 shows some numerical results for the two different archiving strategies and different discretizations. In all cases, $\epsilon = 1$ has been chosen and $N = 100\,000$ randomly chosen points out of the domain $Q = [-2, 2]^2$ have been inserted into the archivers. Figure 4 (a) shows the result of *ArchiveUpdate*$M_{Q,\epsilon}$, where the final archive A_{final} consists of the numerically intractable amount of 16 607 elements. Figure 4 (b) shows a result of the archiver *ArchiveUpdate*$M_{Q,\epsilon}D_x$ using $\Delta = (0.1, 0.1)$ leading to 175 archive entries. Though this is, unlike the first result, a tractable number of elements, similar small values of the entries of Δ can quickly lead to similar problems when increasing the number of parameters. A possible remedy could be (if possible) to assign different values for the entries Δ_i according to their significance. Figure 4 (c) shows a result of *ArchiveUpdate*$M_{Q,\epsilon}D_x$ for $\Delta = (0.1, 1)$. Hereby, it is assumed that a change in x_1 is relatively important (and results with even small changes in x_1 have to be stored) while a change in parameter x_2 is not of relevance (or not as relevant as a change in x_1). Accordingly, the result in Fig. 4 (c) resembles more a 1D set than a 2D set (as it is the case for $M_{Q,\epsilon}$). Proceeding in a similar manner, the 'dimension' of $M_{Q,\epsilon}$ (and hence the number of elements in the



(a) $ArchiveUpdateM_{Q,\epsilon}$, $|A_{final}| = 16{,}607$

(b) $ArchiveUpdateM_{Q,\epsilon}D_x$, $\Delta = (0.1, 0.1)$, $|A_{final}| = 175$

(c) $ArchiveUpdateM_{Q,\epsilon}D_x$, $\Delta = (0.1, 1)$, $|A_{final}| = 19$

Fig. 4 Numerical results for SOP (27) using different archiver and different discretization parameters archive) can be reduced in any order according to the problem and the computational limitations: if, in the extreme case, the value $\Delta_i = b_i - a_i$ is chosen, where a_i and b_i are the bounds for parameter x_i, then the archiver makes no distinction with respect to x_i, and hence, the 'dimension' of the outcome set obtained by $ArchiveUpdateM_{Q,\epsilon}D_x$ is indeed reduced.

5.3 Transfer from earth to apophis

In addition to the previous academic examples, three interplanetary trajectory design problems are considered in the following.

The peculiarity of all problems (as well as other problems related to space mission design) is that: the local minima are – similar to Rosenbrock's famous banana function – typically located in long, narrow valleys, often flat in one particular direction; there are multiple local minima grouped in clusters with a funnel structure [**24**]. Hence, such problems are typically (i) hard to solve and (ii) the approximation of $M_{Q,\epsilon}$ by using $ArchiveUpdateM_{Q,\epsilon}D_x$ can contain a tremendous number of archive entries for small or even moderate values of Δ. To avoid this and to obtain a meaningful approximation of $M_{Q,\epsilon}$, the authors of this article have proceeded as described in the previous subsection: the domain was divided into 'significant' and 'insignificant' parameters. For the significant parameters (launch date, initial velocity, and time of flights), the discretization parameter $\Delta_i = (b_i - a_i)/0.01$, i.e. 1 per cent of the given range $[a_i, b_i]$, has been chosen, and for the insignificant parameters (angles, k_2) the value $\Delta_j = (b_j - a_j)/0.1$ has been chosen.

The first example is an apparently simple transfer from the Earth to the asteroid Apophis. The transfer is performed by applying a change of velocity at departure, or Δv, to leave the Earth, and a change of velocity at Apophis to rendezvous with the asteroid. The cost function is the sum of the modula of the two velocity variations. Due to the similarity of the orbital elements of the two celestial bodies, there exist many local minima corresponding to many possible ways to reach Apophis. The cost function depends on the launch date and transfer time. Here, both parameters have been chosen to be significant and a wide range of values was chosen for both parameters (about 7000 days for the launch date and 800 for the transfer time). Over such a wide range of values, identifying the global minimum is a challenge. Figure 5 shows the level curves of the objective function. Darker areas correspond to lower values of the total Δv. In Fig. 1, one can observe a large number of local minima with the associated long narrow neighbourhood mentioned before. The minima are grouped and the clusters are distributed along the launch-date axis with a certain periodicity. Each cluster, or group of minima, belongs to a different funnel.

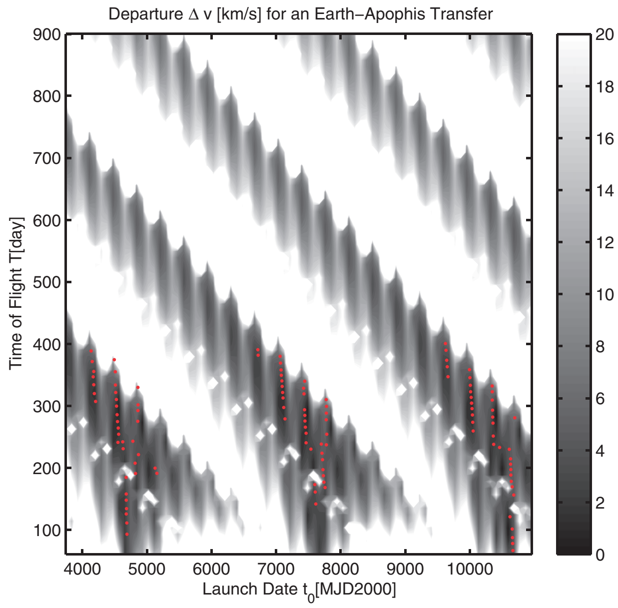


Fig. 5 Numerical results for the objective considered in section 5.3. The figure shows the contour plot of the objective plus the archive entries obtained by the novel algorithm

For this kind of problems, although the identification of the global minimum is useful, it is also not sufficient to design a mission. Decision makers require other two pieces of information: given an optimal launch date and transfer time, alternative launch dates and transfer times with similar cost are required as back up options, for each locally optimal launch date and transfer time all transfer solutions in a close neighbourhood of the local minimum are required. The set of solutions that are in a neighbourhood of the local minimum and at a distance ϵ from it in the image space, form the so-called launch window. A wide launch window means a flexible mission that can accommodate delays and contingencies. The set of local minima with similar cost represents multiple launch opportunities: a mission with multiple launch opportunities offers a higher degree of robustness and flexibility.

The dots in Fig. 5 are the solutions collected by the archiving strategy using $\epsilon = 0.5$ [km/s], after $n = 1e7$ function evaluations of Differential Evolution, with a population size of 100, $F = 0.9$, and $C_R = 1$. The archiving procedure correctly identified the most interesting launch opportunities (lowest Δv) with their associated local neighbourhood. Therefore, in this case, the decision maker is offered with three groups of launch opportunities and for each one multiple launch windows with transfer times ranging from less that 100 days to over 400 days. All the collected solutions have a total Δv at an ϵ distance from the best solution, therefore, they are all admissible. In fact, the value ϵ is easily set *a priori* based on mission constraint on the available Δv budget. Note that the best known solution for this problem is included in the archive.

6 THE ROSETTA CASE

This second case study is a multigravity assist trajectory from the Earth to the comet 67P/Churyumov–Gerasimenko following the gravity assist sequence that was planned for the spacecraft Rosetta: Earth–Earth–Mars–Earth–Earth–Comet. This mission was initially scheduled for launch on an Ariane 5 launcher. However, due to a failure in the previous launch, the mission had to be rescheduled. Rescheduling a mission with such a complex sequence of gravity assist manoeuvres is not an easy matter. Therefore, for this type of mission, it is desirable to generate multiple transfer options since the start of the mission design process. The trajectory model considered here is the one described in references [**1**, **2**]. A deep space maneuver is allowed along the transfer arc from one planet to the other according to the model presented in references [**1**, **24**]. The objective is the sum of all the deep space manoeuvres plus the initial Δv_0 at departure and the final Δv_f to rendezvous with the comet. The search space for this problem has 22 dimensions and cannot be graphically completely represented. An analysis of this search space can be found in reference [**24**]. Even in this case the local minima are grouped in multiple funnels, for each funnel, the analysis in reference [**24**], revealed a high number of local minima irregularly distributed.

Figure 6 shows three projections of the final archive A_{final} of one run of the algorithm described in section 3 for $\epsilon = 0.5$ [km/s] and Δ as described above. A_{final} consists of a total of 122 elements and contains an approximation of the best known solution 20 P_1 with $f(P_1) \approx 1.34$ [km/s] [**2**] as well as other ϵ-approximate solutions of P_1 within three connected components. The three local optima within the components are shown in Table 5. The clusters in Figure 6 correspond to the funnel structures identified in reference [**24**]. As already mentioned in section 1, the DM is offered (at least) two more options in addition to the best known trajectory. Also, the number of archive entries is tractable since it does not slow down the computational cost significantly. If, hypothetically, for unified small values of Δ_i three points per coordinate direction and connected component

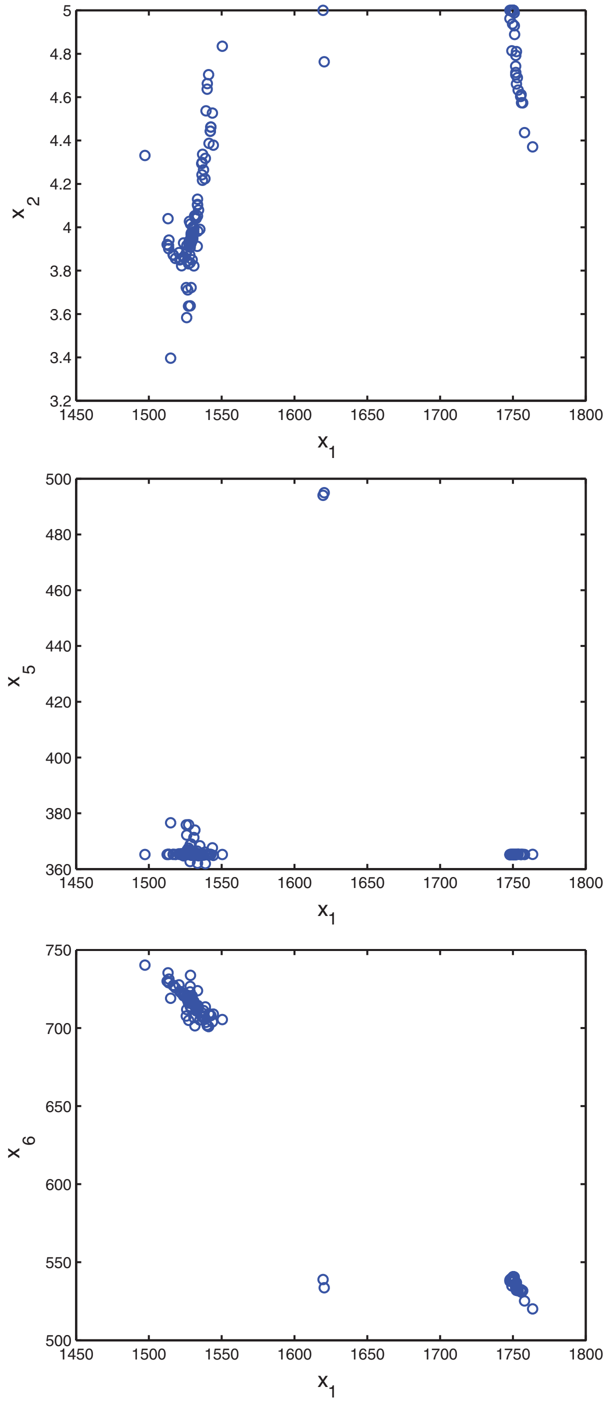


Fig. 6 Numerical results for the Rosetta case. Hereby, x_1 denotes the value of the launch date (MJD2000), x_2 denotes the initial velocity (km/s), x_5 denotes the time of flight for the first arc of the trajectory (d), and x_6 the time of flight for the second arc of the trajectory (d)

would have been required for the approximation (which is much less than shown in Fig. 6), this would have led to a total of $3 \times 3^{22} \approx 10^{11}$ archive entries, which would certainly not have been realizable.

Table 5 The three local solutions P_i, $i = 1, 2, 3$, from the three connected components shown in Fig. 6

Variable	Units	P_1	P_2	P_3
x_1	MJD2000	1.542E+03	1.748E+03	1.620E+03
x_2	km/s	4.443E+00	5.000E+00	5.000E+00
x_3	n/a	9.881E-01	5.146E-01	9.613E-01
x_4	n/a	5.623E-01	2.958E-01	5.000E-01
x_5	Days	3.652E+02	3.652E+02	4.940E+02
x_6	Days	7.082E+02	5.391E+02	5.389E+02
x_7	Days	2.574E+02	6.810E+02	6.811E+02
x_8	Days	7.304E+02	6.307E+02	6.309E+02
x_9	Days	1.850E+03	1.818E+03	1.813E+03
x_{10}	n/a	3.178E-01	5.496E-01	4.151E-01
x_{11}	n/a	8.097E-01	1.088E-01	9.516E-02
x_{12}	n/a	1.361E-01	4.308E-01	3.963E-01
x_{13}	n/a	6.566E-01	2.713E-01	4.703E-01
x_{14}	n/a	4.375E-01	4.908E-01	4.876E-01
x_{15}	n/a	2.986E+00	2.374E+00	1.699E+00
x_{16}	n/a	1.050E+00	1.050E+00	1.050E+00
x_{17}	n/a	3.202E+00	3.326E+00	3.338E+00
x_{18}	n/a	1.050E+00	1.050E+00	1.050E+00
x_{19}	rad	3.273E+00	3.122E+00	3.361E+00
x_{20}	rad	−2.187E-01	−4.443E-01	−4.423E-01
x_{21}	rad	3.135E+00	2.556E+00	2.560E+00
x_{22}	rad	3.554E+00	3.656E+00	3.656E+00
$F(P)$		1.342E+00	1.763E+00	1.770E+00

7 THE CASSINI CASE

The Cassini case is a multigravity assist trajectory from the Earth to Saturn following the sequence Earth–Venus–Venus–Earth–Jupiter–Saturn (EVVEJS). Even in this case, a deep space manoeuvre is allowed along the transfer arc from one planet to the other according to the model presented in reference [**1**, **24**]. The objective is the sum of all the deep space manoeuvres plus the initial Δv_0 at departure and the final Δv_f at arrival at Saturn. This model reproduces the actual Cassini–Huygens mission that was launched in 1997 and successfully entered into orbit around Saturn in 2004. Unlike the Rosetta case, the current hypothesis from previous analyses is that there is only one principal funnel and that the minima are nested in very narrow valleys. Figure 7 shows a final archive A_{final} (with $|A_{final}| = 635$) obtained from this model using the same values for ϵ and Δ as for the Rosetta case.

From Fig. 7, one can see one main cluster with the solutions distributed in two connected groups. In the $x_1 - x_6$ and $x_1 - x_5$ planes, i.e. launch time, time of flight of the first transfer arc, time of flight of the second transfer arc, one can notice that the solutions are aligned along particular directions, revealing a narrow valley structure. In the $x_1 - x_2$ plane, the solutions are much more scattered, although still two main groups can be identified (for further details on this particular problem, please refer to reference[**24**].

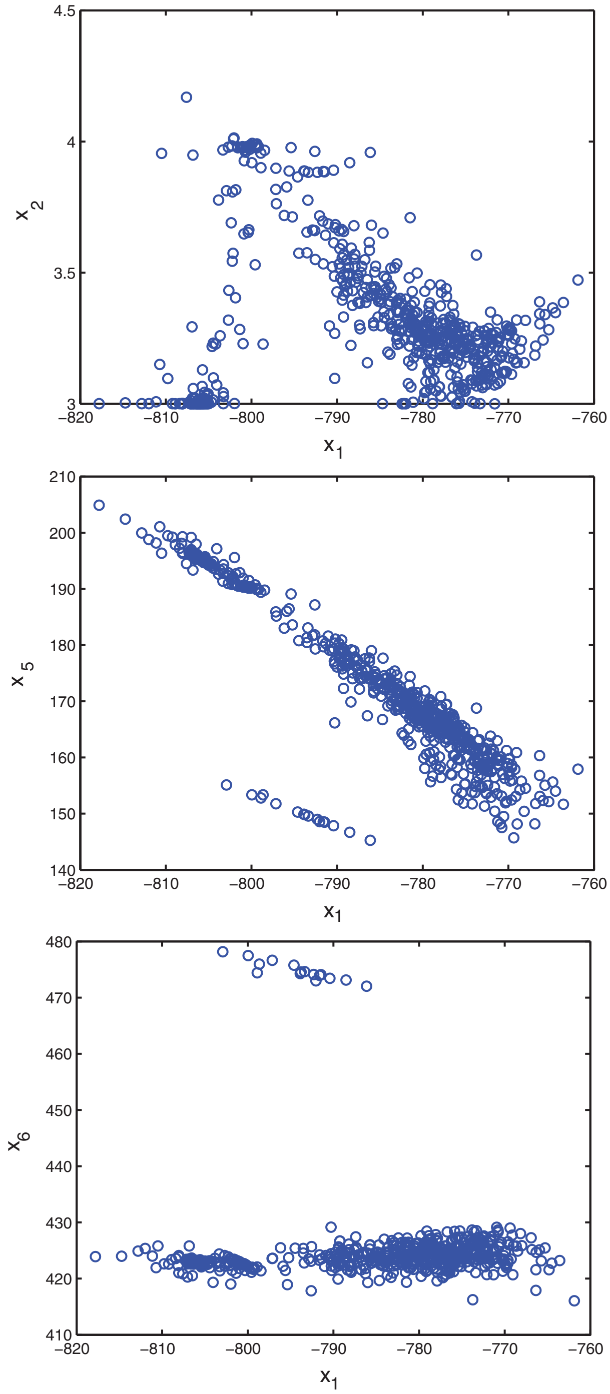


Fig. 7 Numerical results for the Cassini case. Hereby, x_1 denotes the value of the launch date (MJD2000), x_2 denotes the initial velocity (km/s), x_5 denotes the time of flight for the first arc of the trajectory (d), and x_6 the time of flight for the second arc of the trajectory (d)

Therefore, also in this case, the archiving procedure seems to have correctly captured the distribution of the minima, providing information on the structure of the problem. The DM is offered a variety of options which are all admissible, because ϵ is the quantification of the available Δv budget and all differ at least by the value of Δ.

8 CONCLUSIONS AND FUTURE WORK

In this article, the problem of computing the set $M_{Q,\epsilon}$ of ϵ-approximate solutions of a scalar optimization problem with a focus on local minima has been addressed. For this, two archiving strategies have been proposed, one which captures all ϵ-approximate solutions out of the obtained data, and another one which uses a certain discretization strategy. Since the dimension of $M_{Q,\epsilon}$ is typically n, where n is the number of parameters involved in the model, the first archiver is mainly of theoretical interest, and required a suitable discretization. The strategy used in the second archiver is designed to focus on the local minima within $M_{Q,\epsilon}$. However, the outcome of the archiver is crucially dependent on the choice of the discretization parameter $\Delta \in \mathbb{R}^n_+$ which has hence to be chosen problem dependent. Since the 'optimal' choice of this parameter may be *ad hoc* unclear, or intuitive choices may lead to a numerically untractable number of archive entries, one way to reduce the elements in the archive has been discussed which has an analogue effect as the reduction of the dimension of the set of interest and which allows for the efficient treatment of higher dimensional problems. Finally, the efficiency of the search strategy (DE together with the new archiver) has been shown on some benchmark functions and its usefulness has been illustrated showing several models related to space mission design.

As part of our future work, an adaptive choice of Δ would be of particular interest for both theoretical and practical considerations: such an adaptation could for instance be used to explore the neighbourhood of a locally ϵ-approximate solution within $M_{Q,\epsilon}$ since this set is very important to quantify its robustness. Finally, open branches of research can be found when interleaving the archive with the generator heuristic (DE, PSO, etc.) as a matter of feedback into its main population.

FUNDING

C. A. Coello Coello acknowledges support from CONACyT project no. 128554. Adriana Lara acknowledges support from ESFM-IPN and CONACyT to pursue graduate studies at the Computer Science Department of CINVESTAV-IPN. M. Vasile acknowledges support from CONACyT project no. 103570.

REFERENCES

1 **Vasile, M.** and **De Pascale, P.** Preliminary design of multiple gravity-assist trajectories. *J. Spacecraft Rockets*, 2006, **43**(4), 5–22.

2 **Vasile, M., Minisci, E.,** and **Locatelli, M.** A dynamical system perspective on evolutionary heuristics applied to space trajectory optimization problems. In Proceedings of the 2009 IEEE Congress on *Evolutionary computation (CEC 2009)*, Trondheim, 18–21 May 2009, pp. 2340–2347.

3 **Olds, A. D., Kluever, C. A.,** and **Cupples, M. L.** Interplanetary mission design using differential evolution. *J. Spacecraft Rockets*, 2007, **44**(5), 1060–1070.

4 **Peng, L., Wang, Y. Z., Dai, G. M., Chang, Y. M.,** and **Chen, F. J.** Optimization of the Earth-Moon low energy transfer with differential evolution based on uniform design. In Proceedings of the 2010 IEEE Congress on *Evolutionary computation (CEC 2010)*, Barcelona, 18–23 July 2010, pp. 1–8.

5 **Schütze, O., Coello Coello, C. A.,** and **Talbi, E.-G.** Approximating the ϵ-efficient set of an MOP with stochastic search algorithms. In Mexican International Conference on *Artificial intelligence (MICAI 2007)* (Eds A. Gelbukh and A. F. Kuri Morales), Agnascaliontes, Mexico, 4–10 November 2007, pp. 128–138 (Springer-Verlag, Berlin/Heidelberg).

6 **Schütze, O., Coello Coello, C. A., Tantar, E.,** and **Talbi, E.-G.** Computing finite size representations of the set of approximate solutions of an mop with stochastic search algorithms. In Proceedings of the 10th annual conference on *Genetic and evolutionary computation (GECCO 2008)*, Atlanta, Georgia, USA, 12–16 July 2008, pp. 713–720 (ACM, New York, USA).

7 **Schütze, O., Vasile, M.,** and **Coello Coello, C. A.** Approximate solutions in space mission design. In Proceedings of the 10th International Conference on *Parallel problem solving from nature (PPSN 2008)*, Dortmund, Germany, 13–17 September 2008, pp. 805–814.

8 **Schütze, O., Lara, A., Coello Coello, C. A.,** and **Vasile, M.** Computing approximate solutions of scalar optimisation problems and applications in spacemission design. In Proceedings of the 2010 IEEE Congress on *Evolutionary computation (CEC '10)*, Barcelona, Spain, 18–23 July 2010, pp. 1654–1661.

9 **Li, J.-P., Balazs, M. E., Parks, G. T.,** and **Clarkson, P. J.** A species conserving genetic algorithmformultimodal function optimization. *Evolut. Comput.*, 2002, **10**(3), 207–234.

10 **Li, X.** Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization. In Proceedings of the Conference on *Genetic and evolutionary computation (GECCO '04)*, Seatle, USA, 1 June 2004, pp. 105–116 (Springer-Verlag).

11 **Iwamatsu, M.** Locating all the global minima using multi-species particle swarm optimizer: The inertia weight and the constriction factor variants. In Proceedings of the 2006 IEEE Congress

on *Evolutionary computation (CEC '06)*, Vancouver, Canada, 16–21 July 2006, pp. 816–822.
12 **Zhang, J., Huang, D.-S.,** and **Liu, K.-H.** Multi-sub-swarm particle swarm optimization algorithm for multimodal function optimization. In Proceedings of the 2007 IEEE Congress on *Evolutionary computation (CEC '07)*, Singapore, 25–28 September 2007, pp. 3215–3220.
13 **Brits, R., Engelbrecht, A. P.,** and **van den Bergh, F.** Locating multiple optima using particle swarm optimization. *App. Math. Comput.*, 2007, **189**(2), 1859–1883.
14 **Vitela, J. E.** and **Castaños, O.** A real-coded niching memetic algorithm for continuous multimodal function optimization. In IEEE Congress on *Evolutionary computation (CEC 2008)*, Hong Kong, China, 1–6 June, 2008, pp. 2170–2177.
15 **Passaro, A.** and **Starita, A.** Particle swarm optimization for multimodal functions: a clustering approach. *J. Artif. Evol. Appl.*, 2008, **8**(2), 1–15.
16 **Li, X.** Niching without niching parameters: Particle swarm optimization using a ring topology. *IEEE Trans. Evolut. Comput.*, February 2010, **14**(1), 150–169.
17 **Saha, A.** and **Deb, K.** A bi-criterion approach to multimodal optimization: Self-adaptive approach. In *Simulated evolution and learning* (Eds K. Deb *et al.*), vol. 6457, *Lecture Notes in Computer Science*, 2010, pp. 95–104 (Springer, Berlin/Heidelberg).
18 **Vasile, M.** and **Locatelli, M.** A hybrid multiagent approach for global trajectory optimization. *J. Global Optim.*, 2009, **44**(4), 461–479.
19 **Laumanns, M., Thiele, L., Deb, K.,** and **Zitzler, E.** Combining convergence and diversity in evolutionarymultiobjective optimization. *Evolut. Comput.*, 2002, **10**(3), 263–282.
20 **Schütze, O., Coello Coello, C. A., Tantar, E.,** and **Talbi, E.-G.** Computing a finite size representation of the set of approximate solutions of an MOP. INRIA Research Report, Technical Report 6492, 2008.
21 **Schütze, O., Laumanns, M., Coello Coello, C. A., Dellnitz, M.,** and **Talbi, E.-G.** Convergence of stochastic search algorithms to finite size Pareto set approximations. *J. Global Optim.*, 2008, **41**(4), 559–577.
22 **Schütze, O., Laumanns, M., Tantar, E., Coello Coello, C. A.,** and **Talbi, E.-G.** Computing gap free Pareto front approximations with stochastic search algorithms. *Evolut. Comput.*, 2010, **18**(1), 65–96.
23 **Price, K. V., Storn, R. M.,** and **Lampinen, J. A.** *Differential evolution: a practical approach to global optimization*, 2005 (Springer Verlag).
24 **Vasile, M., Minisci, E.,** and **Locatelli, M.** An inflationary differential evolution algorithm for space trajectory optimization. *IEEE Trans. Evolut. Comput.* (to appear).