# Computing Approximate Solutions of Scalar Optimization Problems and Applications in Space Mission Design

Oliver Schütze, Adriana Lara, Carlos A. Coello Coello, and Massimiliano Vasile

***Abstract*— In many applications it can be advantageous for the decision maker to have multiple options available for a possible realization of the project. One way to increase the number of interesting choices is in certain cases to consider in addition to the optimal solution $x^*$ also nearly optimal or approximate solutions which differ in the design space from $x^*$ by a certain value. In this paper we address the efficient computation and discretization of the set $E$ of $\epsilon$-approximate solutions for scalar optimization problems. For this we will suggest two strategies to archive and update the data coming from the generation process of the search procedure, and will use Differential Evolution coupled with the new archivers for the computation of $E$. Finally, we will demonstrate the behavior of the archiver empirically on some academic functions as well as on two models related to space mission design.**

## I. Introduction

One common way to solve a real world engineering problem is by transforming it into an optimization problem and to seek for the (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 ones since by this the decision maker (DM) can be offered 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 we consider the objective shown in Figure 1. In case the DM is willing to accept a deterioration of $\epsilon$, $f$ contains next to the global minimizer $x_1$ also the local minimizer $x_2$ which is such an '$\epsilon$-approximate solution' (i.e., the function values of $f(x_1)$ and $f(x_2)$ differ by less than $\epsilon$). As well, all other points in $[a, b] \cup [c, d]$ are also approximate solutions, however, they are all 'dominated' within their connected components by the solutions $x_1$ and $x_2$ and are possibly too near to them in order to give the DM a significant new alternative to either $x_1$ or $x_2$. 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 in order not to confuse the DM and for sake of an efficient computation (since no superfluous options have to be stored and updated).

As another example we consider the problem of designing an 'optimal' trajectory from Earth to the comet 67P/Churyumov-Gerasimenko (see [10], [12], and also Section V.C of this paper). 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 is 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 $T_0(P_1)$).

Here we address the problem of computing approximate solutions of scalar optimization problems. Since the set $E$ of these $\epsilon$-approximate solutions typically forms an $n$-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 work, we focus on the approximation of the local minima within $E$ and discuss possible discretization strategies in case the objective is 'flat' around a local minimum in $E$ (as this is for instance the case for the 'funnels' in models related to space mission design). For this, we will propose and investigate one archiving strategy which we will combine with Differential Evolution (DE) in order to obtain an efficient algorithm for the approximation of $E$.

The current work can be considered as an 'extension' of previous studies on the computation of approximate solutions for multi-objective optimization problems (MOPs), see $[5], [7], [9]$. 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 we will see later on, a discretization of the set of interest is mandatory, and 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 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 nonobservance of an entire (and large)

The first three authors are with the Departamento de Computación, CINVESTAV-IPN, Av. IPN No. 2508, Col. San Pedro Zacatenco, México, D.F. 07360, MEXICO (email: alara@computacion.cs.cinvestav.mx, {ccoello, schuetze}@cs.cinvestav.mx). The second author is also with the UMI-LAFMIA3175 CNRS at CINVESTAV-IPN. The fourth author is with the Faculty of Engineering, University of Glasgow, University Avenue, Glasgow G12 8QQ, SCOTLAND (email: m.vasile@aero.gla.ac.uk).

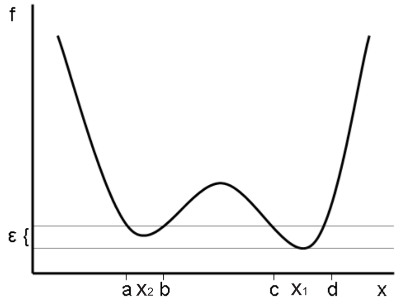



Fig. 1. Example of an objective function with two minima $x_1$ and $x_2$ which are similar in objective space but differ in parameter space.

optimal region. This changes, however, if only one objective is under consideration since in that case the (local or global) optima are typically isolated (as in Figure 1), and hence, a discretization can in principle be performed in parameter space without essential loss of information.
Approximate solutions in space mission design problems have already been considered in [11], where a hybrid multiagent approach has been chosen for their detection. Finally, our approach is similar in spirit to multi-modal optimization, where the aim is to detect all local minima within a given region (e.g., [2], [14]).

The remainder of this paper is organized as follows: in Section II, we give the required background for the understanding of the sequel. In Section III, we present and investigate the set of interest, and propose in Section IV methods for their efficient computation. In Section V, we present some numerical results, and finally draw some conclusions in Section VI.

## II. Background

In the following we consider single-objective optimization problems (SOPs) of the form

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

where $f : Q \subset \mathbb{R}^n \to \mathbb{R}$. For theoretical purposes we will have to assume that the domain $Q$ is compact, the reader may think of an $n$-dimensional box

$$Q = \{x \in \mathbb{R}^n \,:\, a_i \le x_i \le b_i,\, i = 1, \dots, n\}, \tag{2}$$

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

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

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).
Algorithm 1 gives a framework of a generic stochastic optimization algorithm, which will be considered in this work [1]. Here, $Q \subset \mathbb{R}^n$ denotes the domain of the MOP, $P_j$ 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

```
1: P_0 ⊂ Q drawn at random
2: A_0 = ArchiveUpdate(P_0, ∅)
3: for j = 0, 1, 2, ... do
4:     P_{j+1} = Generate(P_j)
5:     A_{j+1} = ArchiveUpdate(P_{j+1}, A_j)
6: end for
```

Finally, we define some distances between points as well as between different sets which we will need to evaluate the approximation quality of our solutions.

*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) $\mathrm{dist}(u, A) := \inf_{v \in A} \|u - v\|$

(b) $\mathrm{dist}(B, A) := \sup_{u \in B} \mathrm{dist}(u, A)$

(c) $d_H(A, B) := \max\{\mathrm{dist}(A, B), \mathrm{dist}(B, A)\}$

## III. The Set of Interest

In the following we define the set of interest, $M_{Q,\epsilon}$, and discuss some of its topological properties.

*Definition 3.1:* Let $\epsilon > 0$, then the set of $\epsilon$-efficient solutions $M_{Q,\epsilon}$ of (1) is defined by

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

We say that a point $x$ is an $\epsilon$-approximate solution of a set $A$ if $f(x) - \epsilon \le f(a)$ for all $a \in A$.

*Example 3.2:* (a) Let $f : \mathbb{R}^n \to \mathbb{R}$ 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

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

i.e., $M_{Q,\epsilon}$ is the closed ball with center 0 and radius $\epsilon^2$.

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

The following little 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 $f$ continuous. Then there exists by continuity of $f$ a neighborhood $N$ of $x^*$ inside $Q$ such that

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

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 (analog to [6]) that in case the objective $f$ is convex, then $M_{Q,\epsilon}$ is connected (and can possibly be computed most efficiently by local search procedures), but this does not hold in general, as the above example shows. Hence, global strategies are typically required for the approximation of $M_{Q,\epsilon}$.
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 $\epsilon$-approximate solution. To be more precise, the set of optima $M_Q$ is contrained in $M_{Q,\epsilon}$ for every $\epsilon > 0$. Furthermore, it is

$$M_Q = \bigcap_{\epsilon>0} M_{Q,\epsilon}. \tag{8}$$

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 (which changes, however, in case the model contains multiple objectives). Regarding this, it is important to note that a discretization of $M_{Q,\epsilon}$ can *not* be performed by looking at the objective values (as e.g. done in [6] for the multi-objective case).

## IV. An Algorithm for the Approximation of $M_{Q,\epsilon}$

In this section we present one possibility to compute approximations of $M_{Q,\epsilon}$. Following the notation of Algorithm 1, we will consider separately the archiver and the generator which form the evolutionary strategy.

### A. Two Archiving Strategies

In the following we discuss two possible archiving strategies aiming for the representation of $M_{Q,\epsilon}$, one which captures all $\epsilon$-approximate solutions out of the obtained data, and one which uses a certain discretization strategy.
The first archiver we consider here, $ArchiveUpdateM_{Q,\epsilon}$, is shown in Algorithm 2. The algorithm captures all efficient solutions out of the obtained data (i.e., the candidate sets $P_i$). To be more precise, let $C_l$ be the set of all considered points up to iteration step $l$, i.e., $C_l := \bigcup_{i=1}^{l} P_i$, then for the archive $A_l$ after the $l$-th step of the search it holds:

$$A_l = M_{C_l,\epsilon} = \{x \in C_l \;:\; f(x) - \epsilon \leq f(y)\; \forall y \in C_l\}. \tag{9}$$

However, due to the dimension of $M_{Q,\epsilon}$ the strategy is apart from the 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 too 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. For this, we propose $ArchiveUpdateM_{Q,\epsilon}D_x$ (see Algorithm 3) 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 neighborhood of these ones in order to obtain a suitable discretization (compare to the motivating example in Section I).
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 $\epsilon$-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 neighborhood $B^\infty_\Delta(p)$ is defined as

$$B^\infty_\Delta(p) := \{x \in \mathbb{R}^n \;:\; |x_i - p_i| < \Delta_i,\; i = 1,..,n\}, \tag{10}$$

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 $\epsilon$-approximate solution of $A$, and (ii) that it is the best point in its neighborhood (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 $\epsilon$-approximate solutions of $p$, and which are not in the $\Delta$-neighborhood 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 we have explicitly stated the best found solution $x_b$. This is in fact not required since the best found solution is always included in the archive due to the construction of Algorithm 3.
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 denote by $x_{b,l}$ the best found solution in

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

**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
```

step $l$, then it holds

$$x_{b,l} \in M_{C_l}, \quad A_l \in M_{C_l,\epsilon}, \tag{11}$$

however, further approximation qualities for finite candidate solutions $\{p_1, \dots, p_s\}, s \in \mathbb{N}$, cannot be given since the final archive $A_l$ depends on the order the candidate solutions $p_i$ are considered. To elucidate the behavior of the distribution of the sequence of archives within $M_{Q,\epsilon}$ convergence analysis is required which we leave for future work. Instead, we will present some numerical results in order to demonstrate the usefulness of the novel strategy, which we will do in the next section.

**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 $x_b$

**if** $f(p) < f(x_{b,0})$ **then**
  $x_b := p$
**else**
  $x_b := x_{b,0}$
**end if**
**if** $f(x_b) + \epsilon < f(p)$ or ($\exists a \in A_0 : p \in B^\infty_\Delta(a_j)$ and $f(a) \leq f(p)$) **then**
  $A := A_o$ ▷ discard $p$
  **return**
**end if**
$A := \{p\}$
**for all** $a \in A_0$ **do**
  **if** $f(a) \leq f(x_b) + \epsilon$ and $a \notin B^\infty_\Delta(p)$ **then**
    $A := A \cup \{a\}$
  **end if**
**end for**

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

$$A \cap B^\infty_\Delta(a) = \{a\}, \tag{12}$$

and hence, the choice of $\Delta$ has a direct influence on the distribution of the archive entries (see e.g. the numerical results in Section V.B). 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. Larger entries of $\Delta$ lead to the focus (and in the ideal case also to a complete reduction) of the local minima within $M_{Q,\epsilon}$, however, the possibility increases that several minima are located within one $\Delta$-neighborhood.
In case the objective $f$ is derived from a real world problem, a rule of thumb could be to choose the entries of $\Delta$ such that two solutions $x_1$ and $x_2$ within the same set $B^\infty_\Delta(x)$ 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 one week), the two trajectories can not be regarded as different (at least according to $T_0$), and the choice would always be in favor 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 behavior of $f$ and the preferences of the DM.

### B. A Possible Generator

For the approximation of $M_{Q,\epsilon}$ we have chosen to use Differential Evolution (DE, see [4]) as the basis for our strategy since this state of the art heuristic has shown its efficiency on a variety of scalar optimization problems, including problems related to space mission design (e.g., [3]). In order to obtain a search procedure aiming for an approximation of $M_{Q,\epsilon}$ instead of 'just' the best solution we utilize—using the notation of Algorithm 1—DE as generator coupled with the new archivers. That is, during the run of DE, we take the members of the population after each generation and insert them into the archiver (see Algorithm 4).

**Algorithm 4** DE + $ArchiveUpdateM_{Q,\epsilon}D_x$

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

## V. Numerical Results

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

### A. Example A

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

$$f(x) = \begin{cases} -\sin(x_1)\sin(x_2) & \text{if} \quad (x_1, x_2) \in [0, 10]^2 \\ -\sin(x_1)\sin(x_2) + 1 & \text{otherwise.} \end{cases} \tag{13}$$

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

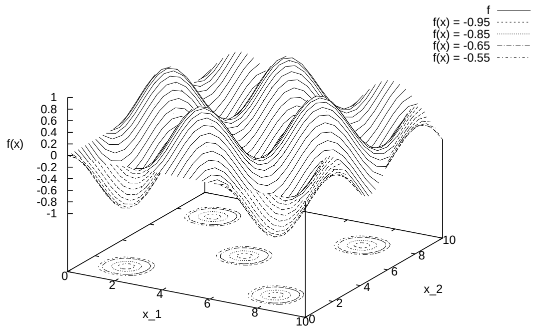



Fig. 2. Surface and contour plot of objective (13) within the ranges $[0, 10]^2$ and the sets $M_{Q,\epsilon}$ for different values of $\epsilon$ (the circles around the minimizers $x_i^*$ indicate the boundaries of $M_{Q,\epsilon}$).

$$x_1^* = \left(\frac{\pi}{2}, \frac{\pi}{2}\right) \quad x_2^* = \left(\frac{\pi}{2}, \frac{5\pi}{2}\right) \quad x_3^* = \left(\frac{5\pi}{2}, \frac{\pi}{2}\right) \\ x_4^* = \left(\frac{3\pi}{2}, \frac{3\pi}{2}\right) \qquad x_5^* = \left(\frac{5\pi}{2}, \frac{5\pi}{2}\right) \tag{14}$$

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 Figure 2).

In order to compare the result of our approach (i.e., DE + archiver) we have chosen a multistart optimization process (using FMINCON of Matlab[1]) and a random search procedure, both equipped with the archiver $ArchiveUpdateM_{Q,\epsilon}D_x$. Tables I to III show some averaged numerical results using the three algorithms and a budget of 12,000 function calls per run. For DE, we have used a population size of 200, the rand/1 strategy, and the $F_{weight}$ factor of the DE was set to 0.9 in all cases. Table I 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 the introduction. Tables II and III are dedicated to the (local) convergence behavior of the archive entries. Since $M_Q$ consists of 5 different solutions, we have chosen to use for a comparison the values (note that both sets $A_{final}$ and $M_Q$ are finite, and hence, we can use min and max)

$$dist(A_{final}, M_Q) = \max_{a \in A_{final}} \min_{i=1,..,5} \|a - x_i^*\|, \tag{15}$$

i.e., the maximal distance from an archive entry of $A_{final}$ to $M_Q$, and the Hausdorff distance

[1]http://www.mathworks.com

$$d_H(A_{final}, M_Q) = \max(dist(A_{final}, M_Q), dist(M_Q, A_{final})), \tag{16}$$

where

$$dist(M_Q, A_{final}) = \max_{i=1,..,5} \min_{a \in A_{final}} \|x_i^* - a\|. \tag{17}$$

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 I.

Summarizing, it can be said that the new strategy (DE + $ArchiveUpdateM_{Q,\epsilon}D_x$) 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 highly depends on the choice of $\epsilon$ and $\Delta$ which is ad hoc unclear for this (and other) academic model.

TABLE I
NUMBER OF COMPONENTS FOUND BY EACH METHOD. MINIMUM, MAXIMUM AND AVERAGE VALUES ARE OVER 100 RUNS.

| Method | Num. of components found | | |
|---|---|---|---|
| | Min | Mean | Max |
| Random Search | 1 | 2.92 | **5** |
| Multistart (fmincon) | 0 | 1.79 | 4 |
| Using DE | **4** | **4.97** | **5** |

TABLE II
DISTANCE FROM THE ARCHIVE OBTAINED WITH EACH METHOD TO THE OPTIMA SET. MINIMUM, MAXIMUM AND AVERAGE VALUES ARE OVER 100 RUNS WITH AT LEAST ONE COMPONENT REACHED.

| 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** |

TABLE III
HAUSDORFF DISTANCE BETWEEN THE ARCHIVE OBTAINED WITH EACH METHOD AND THE OPTIMA SET. MINIMUM, MAXIMUM AND AVERAGE VALUES ARE OVER 100 RUNS WITH AT LEAST ONE COMPONENT REACHED.

| 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** |

### B. Example B

The next academic function we consider is (compare to Example 3.2)

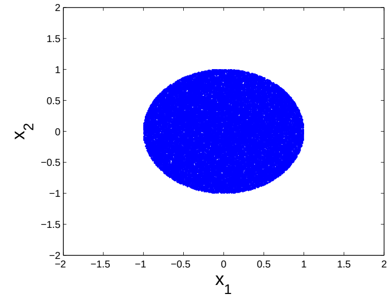
$$f : \mathbb{R}^2 \to \mathbb{R} \\ f(x) = x_1^2 + x_2^2 \tag{18}$$

Figure 3 shows some numerical results for the two different archiving strategies and different discretizations. In all cases we have chosen $\epsilon = 1$ and have inserted $N = 100,000$ randomly chosen points out of the domain $Q = [-2,2]^2$ into the archivers. Figure 3 (a) shows the result of $ArchiveUpdateM_{Q,\epsilon}$, where the final archive $A_{final}$ consists of the numerically intractable amount of 16,607 elements. Figure 3 (b) shows a result of the archiver $ArchiveUpdateM_{Q,\epsilon}D_x$ 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 $\Delta$ 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 $\Delta_i$ according to their significance. Figure 3 (c) shows a result of $ArchiveUpdateM_{Q,\epsilon}D_x$ 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 $x_1$). Hence, the result in Figure 3 (c) resembles rather a 1-dimensional set than a 2-dimensional 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 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.

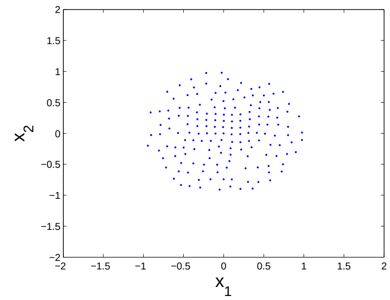
### C. Rosetta

Next to the previous academic examples we consider two interplanetary trajectory design models. The peculiarity of both problems (as well as other problems related to space mission design) is that the local minima are—similar Rosenbrock's famous banana function—typically located in long, narrow, and flat valleys. 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 $\Delta$ (as observed in the two cases at hand since in both cases the dimension of the parameter space is $n = 22$). To avoid this and to obtain a meaningful approximation of $M_{Q,\epsilon}$ we have proceeded as described in the previous subsection: we have divided the domain into 'significant' and 'insignificant' parameters. For the significant parameters (launch date, initial velocity, time of flights) we have chosen the discretization parameter $\Delta_i = (b_i - a_i)/0.01$, i.e., one percent of the given range $[a_i, b_i]$, and for the insignificant parameters (angles, $k_2$) we have chosen the value $\Delta_j = (b_j - a_j)/0.1$.

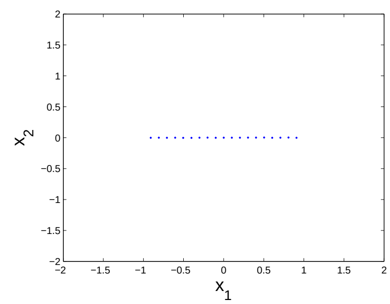
The Rosetta case is a multi gravity assist trajectory from the Earth to the comet 67P/Churyumov-Gerasimenko following the gravity assist sequence that was planned for the spacecraft Rosetta: Earth–Earth–Mars–Earth–Earth–Comet.




(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. 3. Numerical results for SOP (18) using different archiver and different discretization parameters.

The trajectory model we consider here is the one described in [10], [12]. Figure 4 shows three projections of the final archive $A_{final}$ of one run of the algorithm described in Section III for $\epsilon = 0.5$ [km/s] and $\Delta$ as described above. $A_{final}$ consists of a total of 122 elements and contains an approximation of the best known solution $P_1$ with $f(P_1) \approx 1.34$ [km/s] [12] as well as other $\epsilon$-approximate solutions of $P_1$ within three connected components. The three local optima within the components are shown in Table IV. As already mentioned in the Introduction, 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 $\Delta_i$ three points per coordinate direction and connected component would have been required for the approximation (which is much less than shown in Figure 4), this would have led to a total of $3 * 3^{22} \approx 10^{11}$ archive entries, which would certainly not have been realizable.

TABLE IV

THE THREE LOCAL SOLUTIONS $P_i$, $i = 1, 2, 3$, FROM THE THREE CONNECTED COMPONENTS SHOWN IN FIGURE 4.

| Variable | Units | $P_1$ | $P_2$ | $P_3$ |
|---|---|---|---|---|
| $x_1$ | MJD2000 | 1.542E+03 | 1.748E+03 | 1.620E+03 |
| $x_2$ | km/sec | 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-02 |
| $x_{14}$ | 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 |

### D. Cassini

The Cassini case is a multi gravity assist trajectory from the Earth to Saturn following the sequence Earth-Venus-Venus-Earth-Jupiter-Saturn (EVVEJS), but a deep space maneuver is allowed along the transfer arc from one planet to the other according to the model presented in [10], [13]. The objective is in this case the $\Delta v_f$ which is the modulus of the vector difference between the velocity of Saturn at arrival and the velocity of the spacecraft at the same time.

Figure 5 shows a final archive $A_{final}$ (with $|A_{final}| = 635$) obtained from this model using the same values for $\epsilon$ and $\Delta$ as for the Rosetta case. Also here, the DM is offered a variety of options which all differ at least by the value of $\Delta$, and are hence all indeed distinct solutions for the DM.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper we have addressed the problem of computing the set $M_{Q,\epsilon}$ of $\epsilon$-approximate solutions of a scalar optimization problem which includes the detection of multiple minimizers. For this, we have proposed two archiving strategies, one which captures all $\epsilon$-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 a suitable discretization is mandatory. The strategy we use 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, we have indicated one way to reduce the elements in the archive which has an analog effect as the reduction of the dimension of the set of interest and which allows for the efficient treatment of higher dimensional problems. Finally, we have shown the efficiency of the search strategy (DE coupled with the new archiver) on some benchmark functions and its usefulness on two models related to space mission design.

For the future, it remains to consider the limit behavior with respect to convergence and distribution of the sequence of archives under certain assumptions on the generation process (as e.g. done in [5], [8], [7] for multi-objective optimization problems). Further, an adaptive choice of $\Delta$ would be of particular interest for both theoretical and practical considerations: such an adaptation could for instance be used to explore the neighborhood of a locally $\epsilon$-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.


## ACKNOWLEDGEMENTS

The second author acknowledges support from CONACyT to pursue graduate studies at the Computer Science Department of CINVESTAV-IPN. The third author acknowledges support from CONACyT project no. 103570.


## REFERENCES


[1] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, 10(3):263–282, 2002.

[2] J.-P. Li, M. E. Balazs, G. T. Parks, and P. J. Clarkson. A species conserving genetic algorithm for multimodal function optimization. *Evolutionary Computation*, 10(3):207–234, 2002.

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

[4] K. V. Price, R. M. Storn, and J. A. Lampinen. *Differential evolution: a practical approach to global optimization*. Springer Verlag, 2005.

[5] O. Schütze, C. A. Coello Coello, and E.-G. Talbi. Approximating the $\epsilon$-efficient set of an MOP with stochastic search algorithms. In A. Gelbukh and A. F. Kuri Morales, editors, *MICAI 2007: Mexican International Conference on Artificial Intelligence*, pages 128–138. Springer-Verlag Berlin Heidelberg, 2007.

[6] O. Schütze, C. A. Coello Coello, E. Tantar, and E.-G. Talbi. Computing a Finite Size Representation of the Set of Approximate Solutions of an MOP. Technical Report 6492, INRIA Research Report, 2008.

[7] O. Schütze, C. A. Coello Coello, E. Tantar, and E.-G. Talbi. Computing finite size representations of the set of approximate solutions of an mop with stochastic search algorithms. In *GECCO 2008: Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 713–720, New York, NY, USA, 2008. ACM.

[8] O. Schütze, M. Laumanns, C. A. Coello Coello, M. Dellnitz, and E.-G. Talbi. Convergence of stochastic search algorithms to finite size Pareto set approximations. *Journal of Global Optimization*, 41(4):559–577, 2008.

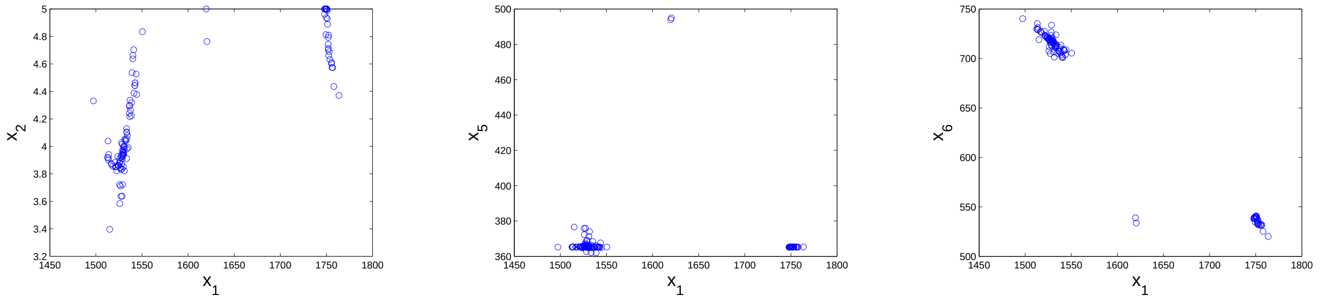



Fig. 4. Numerical results for the Rosetta case. Hereby, $x_1$ denotes the value of the launch date (MJD2000), $x_2$ denotes the initial velocity (km/s), $x_5$ denotes the time of flight for the first arc of the trajectory (d), and $x_6$ the time of flight for the second arc of the trajectory (d).

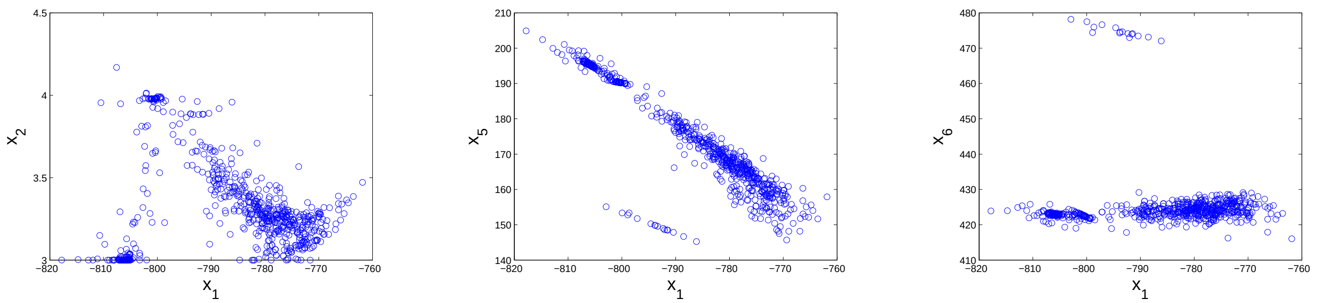



Fig. 5. Numerical results for the Cassini case. Hereby, $x_1$ denotes the value of the launch date (MJD2000), $x_2$ denotes the initial velocity (km/s), $x_5$ denotes the time of flight for the first arc of the trajectory (d), and $x_6$ the time of flight for the second arc of the trajectory (d).


[9] O. Schütze, M. Vasile, and C. A. Coello Coello. Approximate solutions in space mission design. In *PPSN 2008: Proceedings of the 10th International Conference on Parallel Problem Solving From Nature*, pages 805–814, 2008.

[10] M. Vasile and P De Pascale. Preliminary design of multiple gravity-assist trajectories. *Journal of Spacecraft and Rockets*, 43(4):5–22, July–August 2006.

[11] M. Vasile and M. Locatelli. A hybrid multiagent approach for global trajectory optimization. *Journal of Global Optimization*, 44(4):461–479, 2009.

[12] M. Vasile, E. Minisci, and M. Locatelli. A dynamical system perspective on evolutionary heuristics applied to space trajectory optimization problems. In *CEC 2009: Proceedings of the 2009 IEEE Congress on Evolutionary Computation*, pages 2340–2347, 2009.

[13] M. Vasile, E. Minisci, and M. Locatelli. Analysis of some global optimization algorithms for space trajectory design. *Journal of Spacecraft and Rockets*, 2010. (in press).

[14] J. E. Vitela and O. Castaños. A real-coded niching memetic algorithm for continuous multimodal function optimization. In *CEC 2008: IEEE Congress on Evolutionary Computation*, pages 2170–2177, 2008.