


Article

Numerical Computation of Lightly Multi-Objective Robust Optimal Solutions by Means of Generalized Cell Mapping

Carlos Ignacio Hernández Castellanos ^{1,*}, Oliver Schütze ¹, Jian-Qiao Sun ²,
Guillermo Morales-Luna ¹  and Sina Ober-Blöbaum ³

¹ Department of Computer Science, CINVESTAV-IPN, Av. IPN 2508, Gustavo A. Madero, San Pedro Zacatenco, Mexico City 07360, Mexico; schuetze@cs.cinvestav.mx (O.S.); gmorales@cs.cinvestav.mx (G.M.-L.)

² School of Engineering, University of California Merced, Merced, CA 95343, USA; jqsun3@ucmerced.edu

³ Faculty of Computer Science, Electrical Engineering and Mathematics, University of Paderborn, 33098 Paderborn, Germany; sinaober@math.uni-paderborn.de

* Correspondence: chernandez@computacion.cs.cinvestav.mx

Received: 19 September 2020; Accepted: 26 October 2020; Published: 5 November 2020



Abstract: In this paper, we present a novel algorithm for the computation of lightly robust optimal solutions for multi-objective optimization problems. To this end, we adapt the generalized cell mapping, originally designed for the global analysis of dynamical systems, to the current context. This is the first time that a set-based method is developed for such kinds of problems. We demonstrate the strength of the novel algorithms on several benchmark problems as well as on one feed-back control design problem where the objectives are given by the peak time, the overshoot, and the absolute tracking error for the linear control system, which has a control time delay. The numerical results indicate that the new algorithm is well-suited for the reliable treatment of low dimensional problems.

Keywords: cell mapping techniques; multi-objective optimization; robustness

1. Introduction

In many real-world engineering problems, one is faced with the problem that several objectives have to be optimized concurrently leading to a multi-objective optimization problem (MOP). The typical goal for such problems is to identify the set of optimal solutions (the so-called Pareto set) and its image in objective space, the Pareto front. However, in practice, the decision-maker may not always be interested in the best solutions, in particular, if these solutions are sensitive to perturbations [1,2]. In such cases, there exists an additional challenge. One has to search not only for solutions with a good performance but also for solutions that can be implemented, leading to the so-called robust multi-objective optimization problem (RMOP) [3]. In this context, the notion of robustness is not clear since it relies on the information at hand from the given problem as well as the preferences of the decision-maker. Consequently, there exist multiple definitions of robustness according to different scenarios [3–6]. The interested reader is referred to [7] for a survey of the different definitions of robustness.

Recently, the lightly robust multi-objective optimal solutions were proposed [7]. These solutions are often good candidates for the decision-maker since solutions have to be reliable as well as to yield good performances. In this case, a solution is considered to be feasible if it is “close enough” to an optimal solution. Then, the “most reliable” solutions are chosen with respect to the set-based minmax robust efficiency [3]. Thus, lightly robust optimal solutions yield a similar performance to optimal ones while being more reliable.

Feedback controls are most popular in the industry [8]. A great effort has been made in designing optimal feedback control gains for various applications. In the time domain, for example, the overshoot, rise time or peak time, settling time, and the tracking error are often used to characterize the performance of the closed-loop system. It is well known that the overshoot and peak time are conflicting objectives, meaning that when the overshoot goes down, the peak time goes up, and vice versa. It is thus quite natural to consider the multi-objective feedback control design to minimize the overshoot, peak time, and tracking error at the same time. In such kinds of applications, light robust optimal solutions are of great interest, since one would like that the solutions chosen are possible to be implemented and at the same time one is interested in solutions near optimal values.

In this work, it is argued that cell mapping techniques [9,10] are particularly advantageous for the computation of lightly robust multi-objective optimal solutions in optimal control problems. As these methods allow for a thorough investigation of small dimensional problems [11–14]. The algorithm couples generalized cell mapping (GCM) with subdivision techniques [15,16] to first compute the set of nearly optimal solutions. Then, it computes the worst-case scenarios for each solution found exploiting the information about the basin of attraction already computed by the GCM. Finally, the algorithm keeps the most reliable solutions with respect to the set-based minmax robust efficient. The results show that the algorithm can compute a good approximation of the solution set on several low-dimensional test functions. Another advantage of the proposed algorithm is that it can provide other sets of interest such as optimal or approximate solutions with the same effort, which provides an advantage to the decision-maker who can then select the solution to be implemented. The main contributions of the work are the proposal of a novel global algorithm for lightly robust multi-objective optimal solutions as well as its analysis of both academic and control problems.

The remainder of this paper is organized as follows: Section 2 contains the notations and some background required for the understanding of the paper. Section 3 states the generalized cell mapping for lightly optimal solutions. Section 4 presents numerical results on selected benchmark problems. In Section 5, a control optimal problem with uncertainty is studied. Finally, Section 6 concludes and gives some possible paths for future work.

2. Background

In the following, the basic concepts to understand the sequel are presented.

2.1. Multi-Objective Optimization

This work considers continuous multi-objective optimization problems of the form:

$$\min_{x \in Q} F(x), \quad (1)$$

where F is defined as the vector of the objective functions $F : Q \rightarrow \mathbb{R}^k$, $F(x) = (f_1(x), \dots, f_k(x))^T$. Further, assume that each objective $f_i : Q \rightarrow \mathbb{R}$ is continuously differentiable, however, stress that in practice continuity will be enough. In multi-objective optimization, optimality is defined by the concept of *dominance*.

Definition 1. (a) *kleiner* Let $v, w \in \mathbb{R}^k$. Then the vector v is less than w ($v <_p w$), if $v_i < w_i$ for all $i \in \{1, \dots, k\}$. The relation \leq_p is defined analogously.

(b) $y \in Q$ is dominated by a point $x \in Q$ ($x \prec y$) with respect to (1) if $F(x) \leq_p F(y)$ and $F(x) \neq F(y)$, otherwise y is called non-dominated by x ;

(c) $x \in Q$ is a Pareto point if there is no $y \in Q$ that dominates x ;

(d) $x \in Q$ is weakly Pareto optimal if there exists no $y \in Q$ such that $F(y) <_p F(x)$.

The set P_Q of Pareto points within Q is called the Pareto set and its image $F(P_Q)$ the Pareto front. Typically, both sets form $(k - 1)$ -dimensional objects [17].

In some cases it is worth looking for solutions that are ‘close’ in objective space to the optimal solutions, for instance as backup solutions. This is the so-called set of nearly optimal solutions. This set is defined as:

Definition 2 ([18]). Let $\epsilon = (\epsilon_1, \dots, \epsilon_k) \in \mathbb{R}_+^k$ and $x, y \in Q$.

- (a) x is said to ϵ -dominate y ($x \prec_\epsilon y$) with respect to (1) if $F(x) - \epsilon \leq_p F(y)$ and $F(x) - \epsilon \neq F(y)$;
- (b) x is said to $-\epsilon$ -dominate y ($x \prec_{-\epsilon} y$) with respect to (1) if $F(x) + \epsilon \leq_p F(y)$ and $F(x) + \epsilon \neq F(y)$.

The notion of $-\epsilon$ -dominance can be used to define the set of nearly optimal solutions.

Definition 3 ([18]). Denote by $P_{Q,\epsilon}$ the set of points in $Q \subset \mathbb{R}^n$ that are not $-\epsilon$ -dominated by any other point in Q , i.e.,

$$P_{Q,\epsilon} := \{x \in Q \mid \nexists y \in Q : y \prec_{-\epsilon} x\}. \quad (2)$$

Thus, in the remainder of this work, the aim will be to compute the set $P_{Q,\epsilon}$. To the best of the authors’ knowledge, there exist only a few algorithms that aim to find approximations of $P_{Q,\epsilon}$ [19–22]. Most of the approaches use the *ArchiverUpdate* $P_{Q,\epsilon}$ as the critical component to maintaining a representation of the set of interest.

2.2. Uncertain Multi-Objective Optimization

Next, an uncertain multi-objective optimization problem (UMOP) is defined. Here, assume that uncertainties in the problem formulation are given as scenarios from a known uncertainty set $U \subseteq \mathbb{R}^m$. It is also assumed that $F : Q \times U \rightarrow \mathbb{R}^k$. In particular, this work focuses on uncertainties in the decision variables. According to [1], this is referred to as Type B uncertainties modeled in a deterministic way. Thus, the following definitions are adapted to this context.

Definition 4. An UMOP $P(U) = (P(\delta), \delta \in U)$ is defined as the family of parameterized problems”

$$P(\delta) := \min_{x \in Q} F(x + \delta), \quad (3)$$

where $F : Q \times U \rightarrow \mathbb{R}^k$ and $Q \subseteq \mathbb{R}^n$.

Note that it is not clear what a solution to such a family of problems is. In the following, a concept of robustness is introduced [3].

For a given feasible solution x , the worst case of the objective vector is interpreted as a set, namely the set of efficient solutions to the multi-objective problem of maximizing the objective function over the uncertainty set. Formally, this can be written as follows:

Definition 5 (Robust efficiency (re) [3]). Given an UMOP $P(U)$, a feasible solution $\bar{x} \in Q$ is called set-based minmax robust efficient if there is no $x' \in Q \setminus \{\bar{x}\}$ such that $F_U(x') \subseteq F_U(\bar{x}) - \mathbb{R}_{\leq}^k$, where $F_U(x) = \{F(x, \xi) : \xi \in U\}$ and \mathbb{R}_{\leq}^k represents the dominance cone. The robust counterpart of an uncertain multi-objective optimization problem is the problem of identifying all $x \in Q$ which are re. Thus, the robust counterpart problem can be stated as:

$$\min_{x \in Q} \sup_{\delta \in U} F(x + \delta), \quad (4)$$

where $\sup_{\delta \in U}$ is defined as the set of efficient solutions of the following multi-objective optimization problem:

$$\max_{\delta \in U} F(x + \delta). \quad (5)$$

One of the main criticisms to previous definitions is that it could be over-conservative since it considers the worst case. Thus, solutions with a poor performance in terms of their objective functions

could be selected. As a possible remedy, in [7] the authors extended the notion of lightly robust solutions to the multi-objective context. In this case, given a nominal scenario $\hat{\delta} \in U$, let $Q_{\delta}(\hat{\delta})$ be the set of efficient solutions of $P(\hat{\delta})$. For each efficient solution $\hat{x} \in Q_{\delta}(\hat{\delta})$ to $P(\delta)$ and some given $0 \leq \epsilon \in \mathbb{R}^k$ the authors define the uncertain multi-objective optimization problem $LR(\hat{x}, \epsilon, U) := LR(\hat{x}, \epsilon, \delta), \delta \in U$, as the family of parametrized, deterministic multi-objective optimization problems:

$$\begin{aligned} LR(\hat{x}, \epsilon, \delta) := & \min_{x \in Q} F(x + \delta) \\ \text{s.t. } & F(x, \hat{\delta}) \leq \min_{\hat{x} \in P_Q} F(\hat{x} + \delta) + \epsilon. \end{aligned} \quad (6)$$

Definition 6 (Lightly robust efficiency (lre)). Given an uncertain multi-objective optimization problem $P(U)$ with nominal scenario $\delta \in U$ and some $\epsilon \in \mathbb{R}_{\geq}^k$. Then a solution $\bar{x} \in Q$ is called lightly robust efficient for $P(U)$ w.r.t. ϵ if it is set-based minmax robust efficient for $LR(\hat{x}, \epsilon, U)$ for some $\hat{x} \in Q_{\delta}(\hat{\delta})$.

Thus, the robust counterpart of this uncertain multi-objective optimization problem is the problem of identifying all $x \in Q$ which are lre. The robust counterpart problem can be defined as:

$$\min_{x \in P_{Q, \epsilon}} \sup_{\delta \in U} F(x + \delta). \quad (7)$$

2.3. Cell Mapping Techniques

Cell mapping techniques were first introduced in [9] for the global analysis of nonlinear dynamical systems. They transform classical point-to-point dynamics into a cell-to-cell mapping by discretizing both phase space and the integration time. In particular, the phase space discretization bounds the method to a small number of variables that can be considered (say, $n < 10$), but this global analysis offers in turn much more information than other methods. The cell mapping techniques are particularly advantageous for the thorough investigation of low dimensional problems. Such problems occur, for instance, when optimizing the control gains in optimal control [11,12,23–28]. In the context of multi-objective optimization, this is in particular the extended set of options that can be offered to the decision-maker (DM) after analyzing the model. In [29], the authors adapted the simple cell mapping (SCM) to the multi-objective context. The proposed algorithm is capable of computing the Pareto set/front and the set of approximate solutions. The method is particularly advantageous if there exist several possibilities to obtain the same optimal or nearly optimal performance. It is important to note that the relevant information about all these sets of interest is available after one single run of the algorithm (together with an ex-post analysis of the obtained data). In GCM, a cell z is allowed to have several image cells, being the successors of z unlike the previous studies performed with SCM since only one image is allowed by the method. In GCM, each of the image cells is assigned a fraction of the total transition probability, which is called the transition probability with respect to z .

The transition probabilities can be grouped into a transition probability matrix P of order $N_c \times N_c$, where N_c is the total number of cells. Then the evolution of the system is completely described by:

$$p(n+1) = P \cdot p(n), \quad (8)$$

where p is a probability vector of dimension N_c that represents the probability function of the state. This generalized cell mapping formulation leads to an absorbing Markov chain [30].

In the following, some concepts that are useful for our work are presented (see [9] for more details).

A Markov chain is absorbing if it has at least one absorbing state, and it is possible to go to an absorbing state from every state (not necessarily in one step).

Two types of cells can be distinguished: A periodic cell i is a cell that is visited infinitely often once it has been visited. In our work, the focus is on periodic cells of period 1, i.e., $P_{ii} = 1$. These kinds of cells correspond to the local optima candidates.

A transient cell is by definition a cell that is not periodic. For an absorbing Markov chain, the system will leave the transient cells with probability one and will settle on an absorbing (periodic) cell.

To consider an arbitrary absorbing Markov chain, renumber the states so that the transient states come first. If there are r absorbing states and ts transient states ($N_c = r + ts$), the transition matrix has the following canonical form:

$$P = \begin{pmatrix} I & 0 \\ R & Q \end{pmatrix}, \quad (9)$$

where Q is a ts by ts matrix, R is a nonzero ts by r matrix, 0 is an r by ts zero matrix, and I is the r by r identity matrix. The matrix Q gathers the probabilities of transitioning from some transient state to another whereas matrix R describes the probability of transitioning from some transient state to some absorbing state.

For an absorbing Markov chain the matrix $I - Q$ has an inverse $N = (I - Q)^{-1}$. The (i, j) -entry n_{ij} of the matrix N is the expected number of times the chain is in state s_j , given that it starts in state s_i . The initial state is counted if $i = j$.

$$N = I + \sum_{k=1}^{\infty} Q^k \quad (10)$$

is called the fundamental matrix of the Markov chain.

The absorbing probability is defined as the probability of being absorbed in the absorbing state j when starting from transient state i , which is the (i, j) -entry of the matrix $B = NR$. In terms of cell mapping, the set of all $B_{i,j} \neq 0$ for $i \in [1, \dots, ts]$ is called the basin of attraction of state j , and an absorbing cell within that basin is called the attractor.

Table 1 describes the nomenclature of relevant variables used in the manuscript.

Table 1. Nomenclature of relevant variables used in the manuscript.

Symbol	Description
P_Q	Pareto set
$P_{Q,\epsilon}$	Set of approximate solutions
P	Transition probability matrix
N	Fundamental matrix of the Markov chain
p_{ij}	Transition probability from cell s_i to cell s_j
$N_e(s_i)$	Set of neighboring cells
bc_i	Set of neighboring cells that dominate s_i
pg_i	Set of neighboring cells mutually nondominated s_i
S_B	Set of solutionsc
$B_d(S_B)$	Set of cell collection after d iterations that contains the solution set
Δ_2	Averaged Hausdorff distance with 2-norm

3. Proposed Algorithm

This section presents the algorithm for the computation of lightly robust optimal solutions.

3.1. General Framework

In the following, the general procedure to compute lightly robust optimal solutions is presented (Algorithm 1). First, the algorithm computes GCM to compute the canonical matrix (line 2 of Algorithm 1). The sub-matrix I contains the periodic cells (candidate optimal solutions of the nominal MOP). Next, these solutions are the starting point to look for nearly optimal solutions with a backward search algorithm (line 3 of Algorithm 1). Then, the cells containing the set of nearly optimal solutions are subdivided and the process is repeated for a number of iterations. After that, the algorithm computes the worst case for each cell found in the previous step by solving $\max_{\delta \in U} F(x + \delta)$ where

$x \in P_{Q,\epsilon}$ (line 6 of Algorithm 1). Finally, the algorithm uses an archiver to filter the best sets of worst cases (line 7 of Algorithm 1). The next sections give details on how to perform each of the steps.

Algorithm 1 GCM for Multi-objective Light Robust Optimal Solutions

Require: F : objective function, $\delta \in \mathbb{R}^n$: error, $lb \in \mathbb{R}^n$ and $ub \in \mathbb{R}^n$: lower and upper bounds respectively, $N^0 \in \mathbb{R}^n$: cells per dimension, s^0 set of cells, $iter$ number of subdivision steps
Ensure: LR : Set of lightly robust solutions

```

1: for  $l = 0, \dots, iter$  do
2:    $[P^l, \bar{s}^l] \leftarrow GCM(F, s^l, lb, ub, N^l)$ 
3:    $P_{Q,\epsilon}^l \leftarrow BackwardSearch(P^l, \bar{s}^l, \epsilon)$ 
4:    $[s^{l+1}, N^{l+1}] \leftarrow Subdivide(P_{Q,\epsilon}^l, l+1)$ 
5: end for
6:  $WC \leftarrow ComputeWC(P^{iter}, s^{iter}, \delta)$ 
7:  $LR \leftarrow ArchiveUpdatePre(WC, [])$ 
8: return  $LR$ 

```

3.2. Generalized Cell Mapping for Multi-Objective Optimization

In order to use GCM in the context of multi-objective optimization, one has to define the dynamical system to be used. In the following, the dynamical system used is described and is based on Pareto dominance. In this case, a given cell $s_i \in S$ will map to those neighbors $N_e(s_i)$ that are better according to at least one objective function (i.e., those cells that dominate s_i) (Equation (11)). If no neighbor yields an improvement, then it belongs to a periodic group (candidate to be a local optimum) (Equation (12)). Then, for each neighbor, the algorithm assigns a probability proportional to the improvement in terms of the objective functions (Equation (13)).

$$bc_i = \{s_j | F(s_j) \preceq F(s_i) \text{ for all } s_j \in N_e(s_i)\} \quad (11)$$

$$pg_i = \{s_j | F(s_j) = F(s_i) \text{ for all } s_j \in N_e(s_i)\} \quad (12)$$

$$p_{ij} = \begin{cases} \frac{\|F(s_i) - F(s_j)\|}{\left(\sum_{k=1}^{|bc_i|} \|F(s_i) - F(s_k)\|\right)} & , \text{ if } s_j \in bc_i \\ |pg_i|^{-1} & , \text{ if } bc_i = \emptyset \text{ and } s_j \in pg_i \\ 0 & , \text{ otherwise} \end{cases} \quad (13)$$

Now that a suitable dynamical system has been defined, it is possible to apply GCM to the multi-objective context. Algorithm 2 shows the key elements to compute the global properties of the MOP at hand. For each cell z , $F(z)$ is compared to the objective values of its neighbors $N_e(z)$. Next, a probability is assigned, proportional to their function values, to pass into those cells. If there is no better neighbor cell, the transition probability is divided by the number of neighbors with the same function value and assigned to them. Note that dominated neighbor cells always gains a transition probability of 0. One of the advantages of GCM is its global nature. The methods allow scape saddle points since it analyzes all the search space to look for a promising direction.

Algorithm 2 Generalized Cell Mapping for Optimization

Require: F : objective function, s : set of cells, lb, ub : lower and upper bounds, N : cells per dimension
Ensure: P, \bar{s}

```

1: Compute  $F(s_i)$  for all  $s_i \in N_e(s_i)$ 
2: Compute the set of better cells  $bc$  with Equation (11) for all  $s_i \in N_e(s_i)$ 
3: Compute the set of equal cells  $pg$  with Equation (12) for all  $s_i \in N_e(s_i)$ 
4: Compute probability  $p$  with Equation (13) for all  $s_i \in N_e(s_i)$ 
5: Compute canonical form of as in Equation (9) and rearrange  $s$  into  $\bar{s}$ 
6: Return  $P, \bar{s}$ 

```

3.3. Computing Approximate Solutions with Backward Search

After one run of the GCM algorithm, the algorithm has gathered information on the global dynamics of the system and is able to approximate the set of interest in a post-processing step. For the problem at hand, the archiving technique *ArchiveUpdate* $P_{Q,\epsilon}$ [20,22] is used.

The integration of both algorithms is as follows: Algorithm 3 updates the archive first with the periodic cells discovered with GCM and continues with the rest of the periodic motion by inverting the cell mappings. First, a queue is generated with the periodic cells, and until the queue is empty the algorithm searches for nearly optimal solutions. The algorithm takes advantage of the fact that GCM has already encoded the mappings in the canonical matrix. Thus, it is possible to exploit that information to perform a breadth-first search where new cells are enqueued if they are accepted by the archiver. Note that if a cell s is not accepted by the archiver neither would the cells that map to s , since by construction these cells are dominated by s . Thus, Algorithm 3 computes the set of nearly optimal solutions without testing all cells in the search space.

Algorithm 3 Computation of $P_{Q,\epsilon}$ with backward search

Require: P : canonical form of probability matrix, s : set of cells

Ensure: $P_{Q,\epsilon}$ approximation

```

1:  $A \leftarrow \text{ArchiveUpdate}P_{Q,\epsilon}(I, [], \epsilon)$ 
2: Create a queue  $Q$  using  $A$ 
3: while  $Q \neq \emptyset$  do
4:    $cell \leftarrow Q.dequeue()$ 
5:    $c \leftarrow P_{cell}^T$ 
6:    $A \leftarrow \text{ArchiveUpdate}P_{Q,\epsilon}(c, A, \epsilon)$ 
7:    $Q.enqueue(c \cap A)$ 
8: end while
9: Return  $A$ 
```

3.4. Subdivision

When using GCM all the search space is analyzed. Thus, it is a common need to find a compromise between computational effort and precision. A promissory approach is to use GCM to compute a raw picture of promissory regions. Afterward, with this preliminary result, the algorithm now focuses on these regions to perform a fine search. Algorithm 4 shows the steps to perform the subdivision of the set of cells that contain nearly optimal solutions.

Algorithm 4 Subdivision

Require: $P_{Q,\epsilon}$: $P_{Q,\epsilon}$ approximation, l : subdivision level

Ensure: $\hat{\mathcal{B}}_l$: new collection of cells

```

1:  $\mathcal{B}_0 := \{v \in V : \exists \tilde{v} \in V, \tilde{v} \preceq v\} \subset P(\hat{P}_{Q,\epsilon}, d_0)$ 
2: Construct  $\hat{\mathcal{B}}_l \subset P(\hat{P}_{Q,\epsilon}, d_0 + l)$  from  $\mathcal{B}_{l-1}$  such that  $\bigcup_{B \in \hat{\mathcal{B}}_l} B = \bigcup_{B \in \mathcal{B}_{l-1}} B$ 
3:  $j \leftarrow (l \bmod n)$ 
4:  $N^l \leftarrow N^{l-1}$ 
5:  $N_j^l \leftarrow 2N_j^{l-1}$ 
6: Return  $\hat{\mathcal{B}}_l, N^l$ 
```

To realize the subdivision, multi-level partitions of $\hat{P}_{Q,\epsilon}$ as described in [15] are considered: A n -dimensional cell B (or box) can be expressed as:

$$B = B(c, r) = \{x \in \mathbb{R} : c_i - r_i \leq x_i \leq c_i + r_i, \quad i = 1, \dots, n\}, \quad (14)$$

where $c \in \mathbb{R}^n$ denotes the center and $r \in \mathbb{R}^n$ the box size, respectively. Every cell B can be subdivided with respect to the j -th coordinate. This division leads to two cells $B_-(c^-, \hat{r})$ and $B_+(c^+, \hat{r})$ where:

$$\hat{r}_i = \begin{cases} r_i & \text{for } i \neq j \\ r_i/2 & \text{for } i = j \end{cases} \quad (15)$$

$$\hat{c}_i = \begin{cases} c_i & \text{for } i \neq j \\ c_i/2 & \text{for } i = j \end{cases} \quad (16)$$

Denote by $P(\hat{P}_{Q,\epsilon}, d)$, $d \in \mathbb{N}$, the set of cells obtained after $d \in \mathbb{N}$ subdivision steps starting with $B(c^0, r^0)$, where in each step $i = 1, \dots, d$ the cells are subdivided with respect to the j_i -th coordinate, where j_i is varied cyclically. That is, $j_i = ((i-1) \bmod n) + 1$. Note that for every point $y \in Q$ and every subdivision step d there exists exactly one cell $B = b(y, d) \in P(\hat{P}_{Q,\epsilon}, d)$ with center c and radius r such that $c_i - r_i \leq y_i < c_i + r_i$, $\forall i = 1, \dots, n$. Thus, every set of solutions S_B leads to a (unique) set of cell collections:

$$\mathcal{B}_d(S_B) := \{b(y, d) \in P(\hat{P}_{Q,\epsilon}, d) : y \in S_B\}. \quad (17)$$

3.5. Compute the Worst Cases

Once the algorithm has computed a suitable representation of the set of nearly optimal solutions, it can search for the worst cases for each solution in $P_{Q,\epsilon}$. As before, the information provided from GCM allows us to compute the set of worst cases with post-processing of the data. For a given cell s , the algorithm finds the $2\lceil \frac{\delta_i}{h_i} \rceil$ neighbors for $i = 1, \dots, n$, where h is the size of the cell and δ is the uncertainty. Then, the algorithm computes the set of worst cases. Note that this can be done by transposing the matrix P and then looking for those cells that do not have any image in $\bar{Q} = \{\bar{x} | x_i - \delta_i \leq \bar{x} \leq x_i + \delta_i\}$. Algorithm 5 shows the procedure to compute the worst cases.

Algorithm 5 Computation of worst cases

Require: $P_{Q,\epsilon}$: $P_{Q,\epsilon}$ approximation, p : probability matrix

Ensure: set of worst cases

- 1: **for all** cell $\in P_{Q,\epsilon}$ **do**
 - 2: Select neighbors of cell (*neighbors*)
 - 3: Compute max of *neighbors* (WC)
 - 4: **end for**
 - 5: Return WC
-

3.6. Compute Best Worst Cases

Finally, it is a requirement to filter the solutions to keep the best worst cases. Algorithm 6 extends the archiver *ArchiveUpdateP_Q* to handle families of solution sets. In this case, both P and A_0 are families of sets. Note that line 3 of the algorithm uses set-based dominance instead of classical Pareto dominance.

3.7. Computational Complexity

In this section, the computational time complexity of each of the algorithms presented is discussed with respect to the number of cells to process.

- GCM: All cells are visited once $O(Nc)$ and for each cell, the algorithm computes its neighbors. The neighbors depend on the type of vicinity that one uses. It could be n if one selects orthogonal neighbors or $3^n - 1$ with the full neighborhood. Note that the number of neighbors is in general much lower than the number of cells. Thus, the complexity of GCM is $O(Nc)$;
- BackwardSearch: In the worst case, all cells have to be visited (all cells are nearly optimal solutions). Since a breadth-first search is used, the cells are visited only once. Next, the complexity

- of $ArchiveUpdateP_{Q,\epsilon}$ is $O(Nc)$ since in the worst case all candidate solutions are compared with the solutions in the archiver. Thus, the complexity of BackwardSearch is $O(Nc^2)$;
- Computation of worst cases: In this case, the algorithm has to analyze at most Nc cell to find their worst cases. The size of each grid is of size $2\lceil \frac{\delta_i}{h_i} \rceil + 1$ since their size is given by the number of neighbors. Note that as in GCM, it takes linear time to find the worst cases. Thus, the complexity of this algorithm is $O(\max(\lceil \frac{\delta_i}{h_i} \rceil)Nc)$ for $i = 1, \dots, n$;
 - $ArchiveUpdateP_{re}$: In the worst case, each candidate solution will be formed by $2\lceil \frac{\delta_i}{h_i} \rceil + 1$ solutions. From this follows that each dominance comparison have a complexity of $O(\max(\lceil \frac{\delta_i}{h_i} \rceil))$. Thus, the complexity of the archiver is $O(\max(\lceil \frac{\delta_i}{h_i} \rceil)Nc^2)$ for $i = 1, \dots, n$.

From the above discussion it follows that the total time complexity of the algorithm to compute lr solutions is $O(\max(\lceil \frac{\delta_i}{h_i} \rceil)Nc^2)$. It is also important to notice that the total number of cells is given by $\prod_{i=1}^n N$. If one would like to maintain the precision, the number of cells required will increase exponentially with the number of dimensions. Thus, the complexity with respect to the number of dimensions is $O(\max(N)^n)$. This is the main reason why GCM is restricted to low dimensional problems.

Algorithm 6 $A := ArchiveUpdateP_{re}(P, A_0)$

Require: population P , archive A_0

Ensure: updated archive A

```

1:  $A := A_0$ 
2: for all  $p \in P$  do
3:   if  $\nexists a \in A : a \prec p$  then
4:      $A := A \cup \{p\}$ 
5:   end if
6:   for all  $a \in A$  do
7:     if  $p \prec_{-\epsilon} a$  then
8:        $A := A \setminus \{a\}$ 
9:     end if
10:  end for
11: end for
12: return  $A$ 
```

4. Numerical Results

In the following, the experimental design used to validate the performance of the novel algorithm is presented. The algorithms were implemented in Matlab[®] R2012a and C (connected through mex files). All executions were performed in a desktop computer with Intel[®] Core[™] i7-2600K CPU 3.40 GHz processor and 4 GB of RAM. Here, four academic test problems were used. These problems were proposed for multi-objective optimization without uncertainty. However, the problems present interesting features for light robust optimization:

Deb99 [31] is a bi-objective optimization problem whose global front is highly sensitive to perturbations since it has a small basin of attraction. This problem was modified to move closer to the local front to the global one. Equation (18) shows the definition of the problem.

$$F(x) = \left(x_1, \frac{g(x_2)}{x_1} \right), \text{ where:} \quad (18)$$

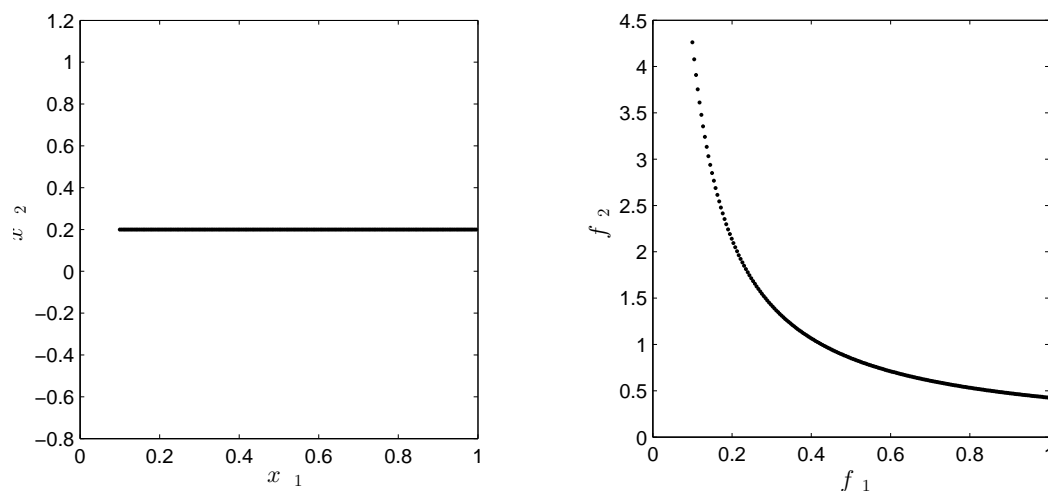
$$g(x) = 2 - \exp \left(- \left(\frac{x_2 - 0.2}{0.004} \right)^2 \right) - 0.8 \exp \left(- \left(\frac{x_2 - 0.6}{0.4} \right)^2 \right)$$

where $0 < x_1 \leq 1$ and $0 \leq x_2 \leq 1$.

Two-on-one [32] was proposed to test the capability of an EMOA to compute equivalent Pareto sets. Sym-part [33], the Pareto set is formed by nine connected components that map to the same region in the Pareto front. SSW [34] is an MOP whose Pareto set falls into four connected components. Due to symmetries of the model two of these components (the two outer curves on the boundary of the domain) map to the same region in the Pareto front.

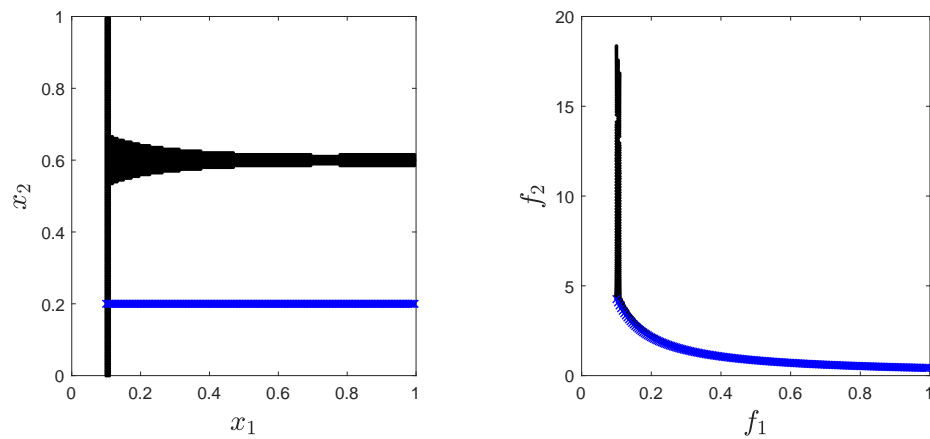
Figure 1 shows the results of the novel algorithm on Deb99 in each step of the algorithm. First, Figure 1a shows the GCM, there it is possible to see that the Pareto set is located in $x_2 = 0.2$. Figure 1b shows the results after the application of the BackwardSearch algorithm. In this case, there are two regions of interest located around $x_2 = 0.2$ and $x_2 = 0.6$. These regions correspond to the local Pareto sets. Then, once the set $P_{Q,\epsilon}$ was computed, the next step is to compute the worst cases for each $x \in P_{Q,\epsilon}$. Figure 1c shows the computation of worst cases. Here, it is possible to observe a curve connecting $[1, 2]$ and $[0.1, 10]$ that did not appear in the set of nearly optimal solutions. This curve corresponds to the worst case of the global Pareto front. Finally, Figure 1d shows the lightly robust optimal solutions after the application of the *Archiver* P_{re} . There, the global Pareto front is dominated (in the re sense) by the local front. Thus, the lightly robust solutions are those solutions in the local front located in $x_2 = 0.6$.

Now, the results for all the problems are shown. Table 2 shows the parameters used to perform the experiments. ϵ denotes the allowed deterioration from the Pareto optimal solutions, δ represents the uncertainty, and N is the number of cells used per dimension. In this experimental study, the Δ_2 is used [35–37] indicator to measure the distance of the best solutions found by the algorithm to the real solution in decision space. Note that this measure can be interpreted as the deviation of the solution of the algorithm to the real solution.

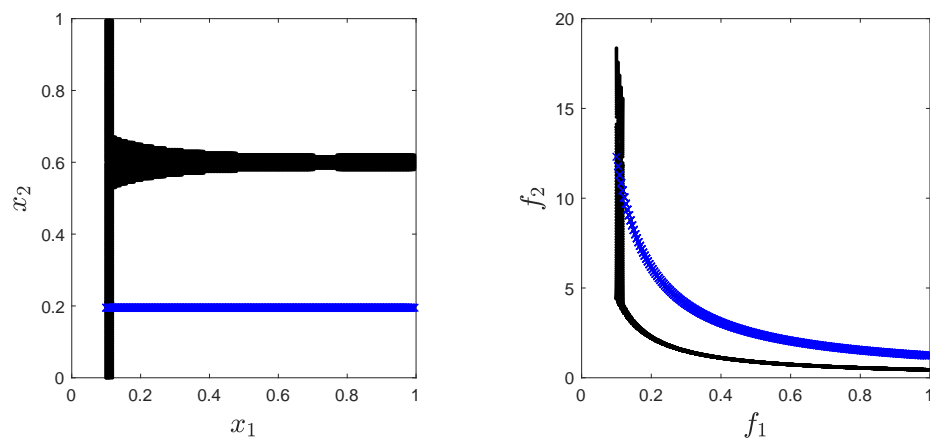
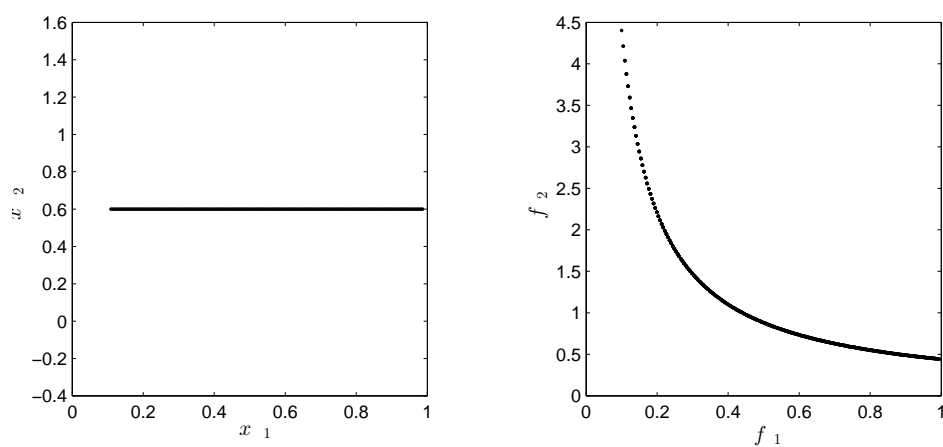


(a) Pareto optimal solutions of Deb99 found by generalized cell mapping (GCM).

Figure 1. Cont.



(b) Nearly optimal solutions of Deb99 after the backward search.

(c) Worst cases of $P_{Q,\epsilon}$ of Deb99.

(d) Lightly robust solutions of Deb99 computed by GCM.

Figure 1. Numerical results on Deb99. In blue, the nominal Pareto set/front. It is possible to observe that due to the uncertainty, the global nominal front deteriorated up to the point that it becomes dominated by the local front.

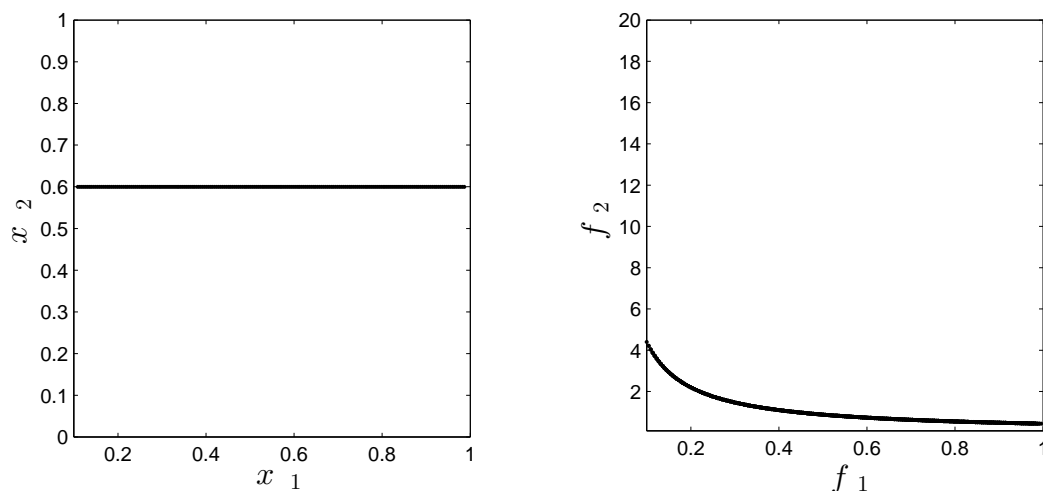
Table 2. Parameters used for each problem.

Problem	ϵ	δ	N
Deb99	$\begin{bmatrix} 0.0110 \\ 0.0110 \end{bmatrix}$	$\begin{bmatrix} 0.0068 \\ 0.0075 \end{bmatrix}$	$\begin{bmatrix} 200 \\ 200 \end{bmatrix}$
Two-on-one	$\begin{bmatrix} 0.1000 \\ 0.1000 \end{bmatrix}$	$\begin{bmatrix} 0.0450 \\ 0.0450 \end{bmatrix}$	$\begin{bmatrix} 200 \\ 200 \end{bmatrix}$
Sym-part	$\begin{bmatrix} 0.1500 \\ 0.1500 \end{bmatrix}$	$\begin{bmatrix} 0.3000 \\ 0.3000 \end{bmatrix}$	$\begin{bmatrix} 200 \\ 200 \end{bmatrix}$
SSW	$\begin{bmatrix} 0.0100 \\ 0.0001 \end{bmatrix}$	$\begin{bmatrix} 3.00 \\ 3.00 \\ 3.00 \end{bmatrix}$	$\begin{bmatrix} 20 \\ 20 \\ 20 \end{bmatrix}$

Figures 2 and 3 show the GCM approximation of the set of lightly robust optimal solutions as well as their respective worst-case images. Up to date, no other method exists that deals with this problem class. In order to appreciate the results and for the sake of comparison, the following possible alternative approach is used through the use of an archiver: A number of uniform random samples are produced. Next, $ArchiverUpdateP_{Q,\epsilon}$ is used to obtain the approximate solutions is applied. Then, for each solution, the algorithm samples in their neighborhood are defined by the uncertainty and find the set of worst cases. Finally, the best-worst cases are computed. To have a fair comparison, the inner sampling was set to 100 solutions and the rest for the sampling in the whole search space. Table 3 shows the Δ_2 values in decision space between the real solution and approximation set found.

Table 3. Δ_2 values and GCM and archiver for each problem.

Problem	GCM	Random
Deb99	0.0015	0.1484 (0.0371)
Two-on-one	0.0124	0.3290 (0.1941)
Sym-part	0.0739	6.3411 (1.3068)
SSW	8.2199	11.7823 (1.0938)

**Figure 2.** Cont.

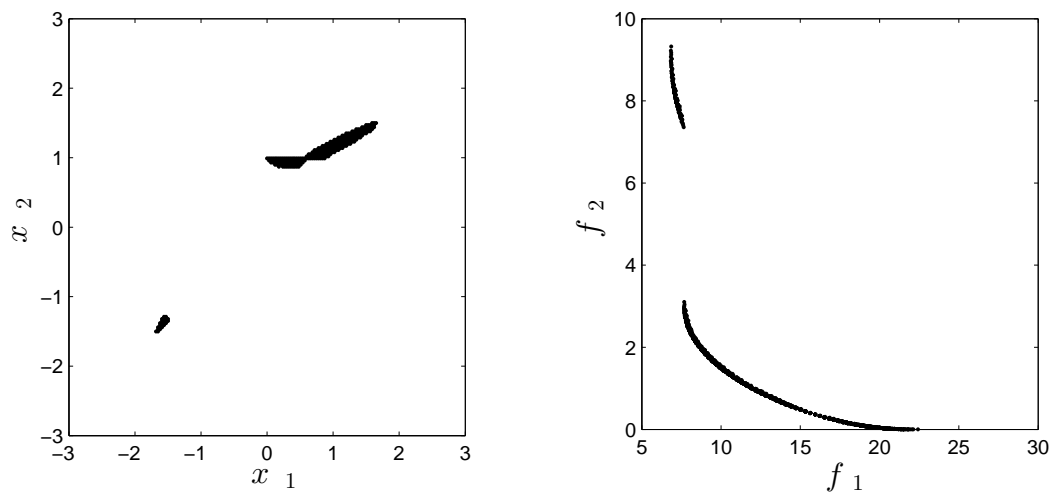


Figure 2. Numerical results of GCM on the academic problems. Decision space (**left**) and objective space (**right**) obtained on the problems Deb99 and two-on-one from above to below.

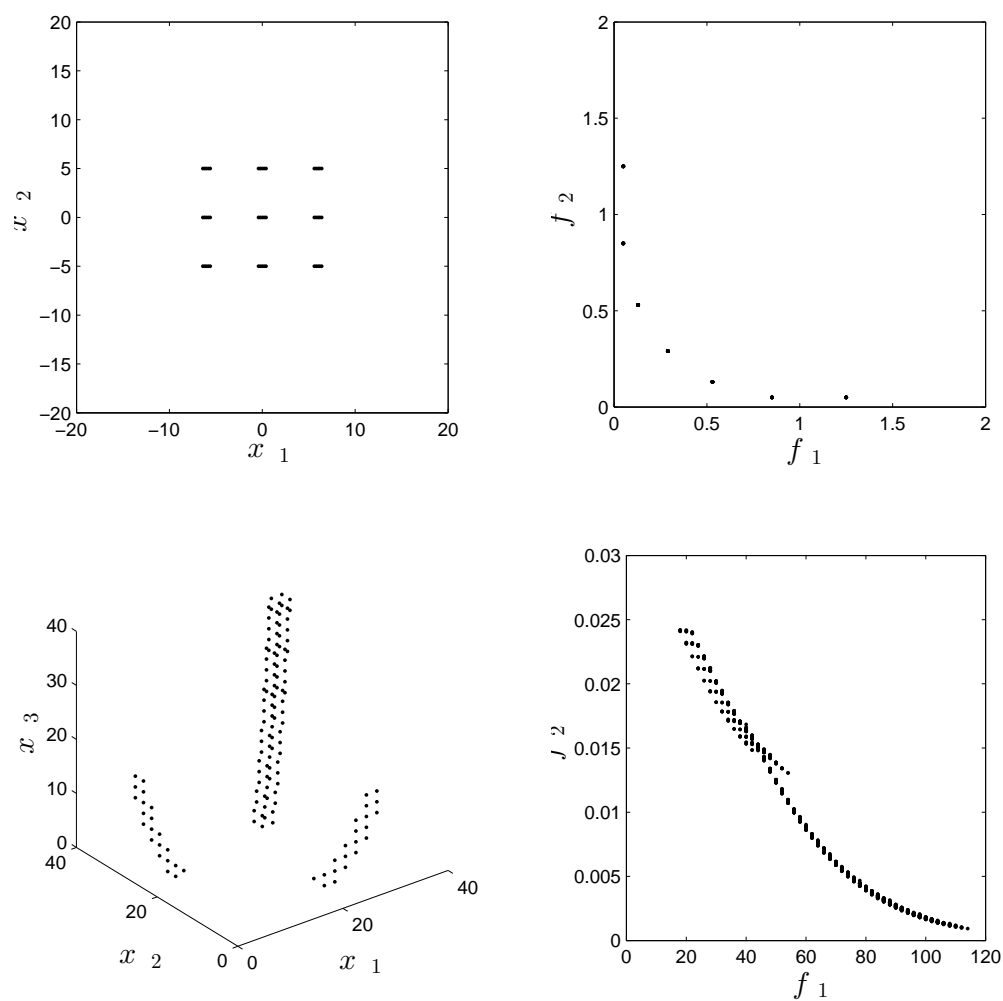


Figure 3. Numerical results of GCM on the academic problems. Decision space (**left**) and objective space (**right**) obtained on the problems sym-part and SSW from above to below.

From the results, it is possible to observe that in Deb99, the solutions in the nominal global front are dominated by those in the local front in terms of lre . In the cases of two-to-one and sym-part, the nominal global front is the lightly robust front. Finally, in SSW one of the connected components that was optimal for the nominal MOP is now dominated in terms of lre . As can be seen, in all cases the GCM approach is superior to the archive-based approach, and the results differ by several orders of magnitude.

5. Application to Optimal Control

Next, a second order oscillator subject to a proportional-integral-derivative (PID) control [12] is considered. Until now, this problem was only studied without uncertainty.

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2x = \omega_n^2u(t), \quad (19)$$

where $\omega_n = 5$, $\zeta = 0.01$,

$$u(t) = k_p[r(t) - x(t)] + k_i \int_0^t [r(\hat{t}) - x(\hat{t})] d\hat{t} - k_d\dot{x}(t), \quad (20)$$

$r(t)$ is a step input, k_p , k_i , and k_d are the PID control gains. The MOP with the control gains $\mathbf{k} = [k_p, k_i, k_d]^T$ as design parameters. The design space for the parameters was chosen and are as follows, $Q = \{\mathbf{k} \in [10, 50] \times [1, 30] \times [1, 2] \subset \mathbb{R}^3\}$.

The multi-objective optimization problem to design the control gain \mathbf{k} is defined as:

$$\min_{\mathbf{k} \in Q} \{t_p, M_p, e_{IAE}\}, \quad (21)$$

where M_p stands for the overshoot of the response to a step reference input, t_p is the corresponding peak time, and e_{IAE} is the integrated absolute tracking error:

$$e_{IAE} = \int_0^{T_{ss}} |r(\hat{t}) - x(\hat{t})| d\hat{t}, \quad (22)$$

where $r(t)$ is a reference input and T_{ss} is the time when the response is close to the steady state. The closed-loop response of the system for each design trial is computed with the help of closed form solutions. The integrated absolute tracking error e_{IAE} is calculated over time with $T_{ss} = 20$ s. In this case, the error was considered to be $\delta = [0.4, 0.29, 0.01]^T$ which corresponds to a 1% error and $\epsilon = [0.10, 0.10, 1]^T$.

GCM was executed with an initial grid of $N = [30, 18, 8]^T$ and 3 subdivision steps. Figure 4 shows the Pareto optimal solutions. Figure 5 shows the nearly optimal solutions. Figure 6 shows the approximation of the lightly robust optimal solutions and their worst-case image found by GCM as well as the Pareto optimal solutions of the nominal problem. Note that optimal solutions and lightly optimal solutions have the same structure. However, there is a $\Delta_2(P_Q, P_{LR}) = 1.0525$. The running time was 709.13 s.

Finally, the response of the system is studied. In [12], the vector $x_{pq} = [40.0, 2.8796, 1.9792]^T$ was selected after the decision making process. In this case, the closest solution in the set of lightly robust optimal solutions is analyzed. This solution is $x_{lr} = [40.5880, 2.7059, 1.9118]^T$ with an euclidian distance of 0.6168 in decision space and 0.9023 in objective space to x_{pq} . Figure 7 shows the response of the system for both solutions. The result shows the trade-off between optimality and robustness. It is possible to observe that x_{lr} deteriorates almost 1% in terms of overshoot. However, this solution is more robust than the solution selected in [12] and thus it would be more desired in practice given the uncertainty.

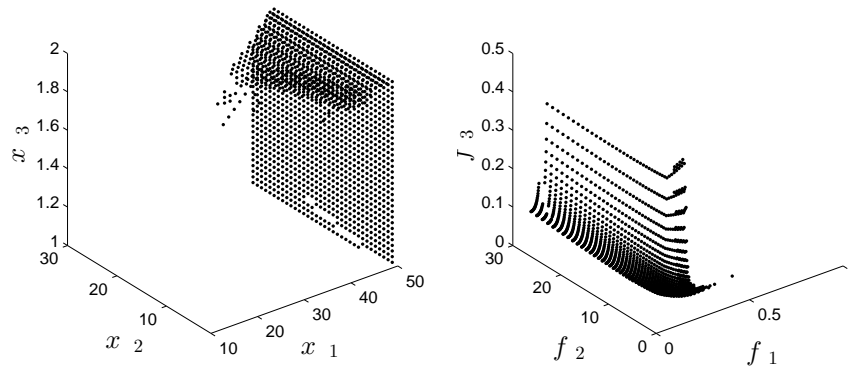


Figure 4. Approximation of the optimal solutions and their worst-case image found by GCM.

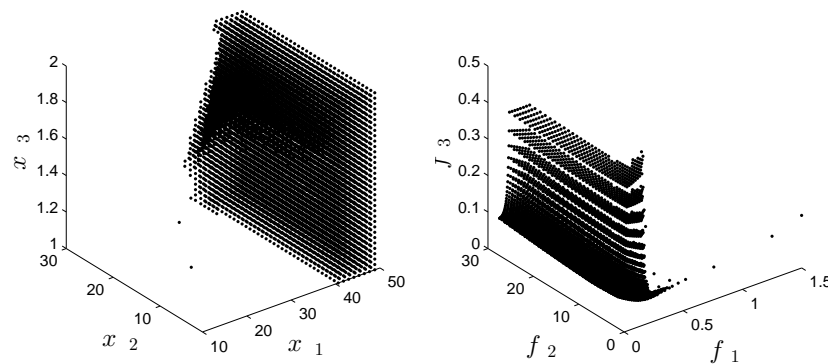


Figure 5. Approximation of the nearly optimal solutions and their worst-case image found by GCM.

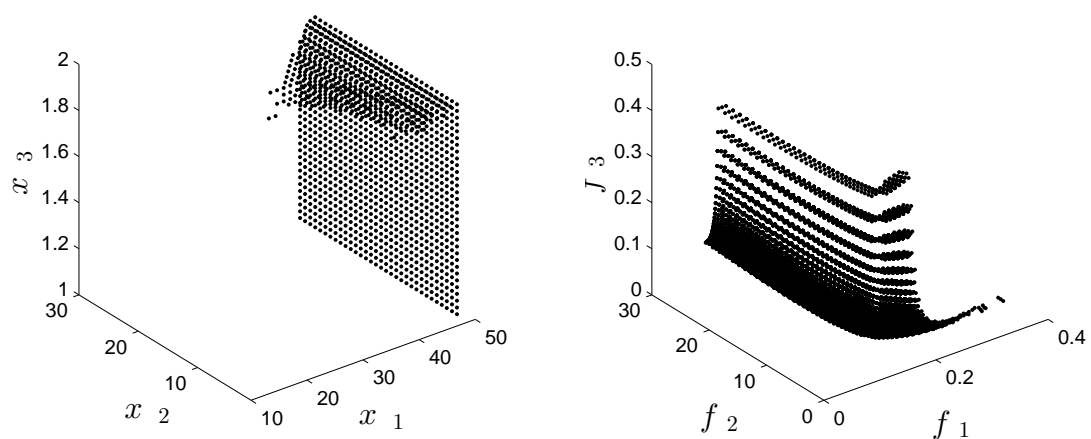


Figure 6. Approximation of the lightly robust optimal solutions and their worst-case image found by GCM.

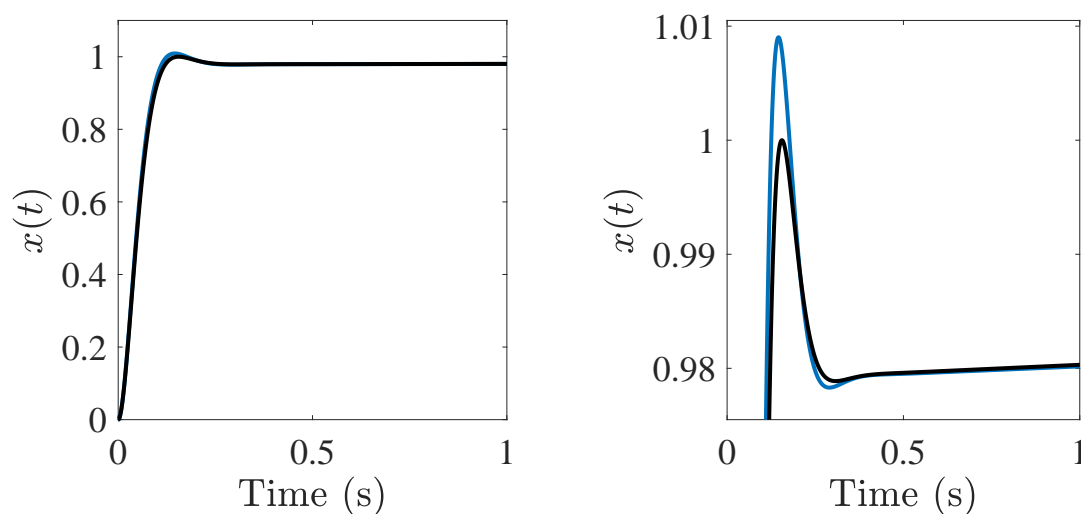


Figure 7. Response of an optimal solution x_{pq} (black) and light robust solution x_{lr} (blue) with respect to time in seconds.

6. Conclusions and Future Work

In this paper, we investigated cell mapping techniques for the numerical treatment of uncertain multi-objective optimization problems in terms of Ire. We adapted the cell mapping techniques to the given context via considering dynamical systems derived from descent methods and argued that the resulting algorithm is particularly beneficial for the thorough investigation of small problems. That is, the new algorithm is capable of detecting solutions that have almost the same performance as the optimal solutions but that are more reliable. This gives the decision-maker solutions that are less susceptible to uncertainties and thus are good candidates to implement. The main advantage of the novel algorithm is that Ir solutions can be computed with the same effort as computing optimal solutions in terms of function evaluations.

Though the results presented in this work are very promising, some points have to be addressed in order to make the algorithm applicable to a broader class of problems. First of all, the main drawback of the cell mapping techniques is that they are restricted to small dimensional problems. Note, however, that the algorithm is highly parallelizable since the core of the algorithm is the mapping of each cell which can be realized with small effort. We expect thus that the use of massive parallelism realized e.g., via GPUs will lead to applicability to higher dimensional problems adapting ideas presented in [38]. Moreover, it would be interesting to study stochastic model predictive control since they are inherently uncertain multi-objective optimizations. Finally, there exist several definitions and kinds of uncertainty. In this work, we focused on Ire and uncertainty in decision variables, such as the one found in manufacturing tolerance errors. Thus, it would be interesting to extend the algorithm to the treatment of other kinds and definitions of uncertainty.

Author Contributions: Conceptualization, C.I.H.C. and O.S.; Data curation, C.I.H.C.; Formal analysis, C.I.H.C., O.S., J.-Q.S. and S.O.-B.; Funding acquisition, G.M.-L. and O.S.; Investigation, C.I.H.C.; Methodology, C.I.H.C., O.S. and J.-Q.S.; Project administration, J.-Q.S. and S.O.-B.; Resources, G.M.-L., O.S., J.-Q.S. and S.O.-B.; Software, C.I.H.C.; Supervision, O.S., J.-Q.S. and S.O.-B.; Validation, C.I.H.C. and G.M.-L.; Visualization, C.I.H.C.; Writing—original draft, C.I.H.C.; Writing—review and editing, C.I.H.C., O.S., J.-Q.S. and S.O.-B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by CONACYT grant numbers 711172 and 285599 and SEP-Cinvestav project no. 231.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Beyer, H.G.; Sendhoff, B. Robust optimization—A comprehensive survey. *Comput. Methods Appl. Mech. Eng.* **2007**, *196*, 3190–3218. [\[CrossRef\]](#)
2. Cuate, O.; Schütze, O. Variation Rate to Maintain Diversity in Decision Space within Multi-Objective Evolutionary Algorithms. *Math. Comput. Appl.* **2019**, *24*, 3.
3. Ehrgott, M.; Ide, J.; Schöbel, A. Minmax robustness for multi-objective optimization problems. *Eur. J. Oper. Res.* **2014**, *239*, 17–31. [\[CrossRef\]](#)
4. Kuroiwa, D.; Lee, G.M. On robust multiobjective optimization. *Vietnam J. Math* **2012**, *40*, 305–317.
5. Doolittle, E.K.; Kerivin, H.L.; Wiecek, M.M. A robust multiobjective optimization problem with application to internet routing. In *Tech. Rep. R2012-11-DKW*; Clemson University Clemson: Clemson, SC, USA, 2012.
6. Fliege, J.; Werner, R. Robust multiobjective optimization & applications in portfolio optimization. *Eur. J. Oper. Res.* **2014**, *234*, 422–433.
7. Ide, J.; Schöbel, A. Robustness for Uncertain Multi-objective Optimization: A Survey and Analysis of Different Concepts. *OR Spectr.* **2016**, *38*, 235–271. [\[CrossRef\]](#)
8. Liu, G.P.; Daley, S. Optimal-tuning nonlinear PID control of hydraulic systems. *Control Eng. Pract.* **2000**, *8*, 1045–1053. [\[CrossRef\]](#)
9. Hsu, C. Cell-to-cell mapping: A method of global analysis for nonlinear systems. In *Applied Mathematical Sciences*; Springer: Cham, Switzerland, 1987.
10. Sun, J.Q.; Xiong, F.R.; Schütze, O.; Hernández, C. *Cell Mapping Methods-Algorithmic Approaches and Applications*; Springer: Cham, Switzerland, 2019.
11. Zufiria, P.J.; Martínez-Marín, T. Improved Optimal Control Methods Based Upon the Adjoining Cell Mapping Technique. *J. Optim. Theory Appl.* **2003**, *118*, 657–680. [\[CrossRef\]](#)
12. Hernández, C.; Naranjani, Y.; Sardahi, Y.; Liang, W.; Schütze, O.; Sun, J.Q. Simple cell mapping method for multi-objective optimal feedback control design. *Int. J. Dyn. Control* **2013**, *1*, 231–238. [\[CrossRef\]](#)
13. Xiong, F.R.; Schütze, O.; Ding, Q.; Sun, J.Q. Finding zeros of nonlinear functions using the hybrid parallel cell mapping method. *Commun. Nonlinear Sci. Numer. Simul.* **2016**, *34*, 23–37. [\[CrossRef\]](#)
14. Gyebroński, G.; Csernák, G. Clustered Simple Cell Mapping: An extension to the Simple Cell Mapping method. *Commun. Nonlinear Sci. Numer. Simul.* **2017**, *42*, 607–622. [\[CrossRef\]](#)
15. Dellnitz, M.; Hohmann, A. A subdivision algorithm for the computation of unstable manifolds and global attractors. *Numer. Math.* **1997**, *75*, 293–317. [\[CrossRef\]](#)
16. Dellnitz, M.; Schütze, O.; Hestermeyer, T. Covering Pareto Sets by Multilevel Subdivision Techniques. *J. Optim. Theory Appl.* **2005**, *124*, 113–155. [\[CrossRef\]](#)
17. Hillermeier, C. *Nonlinear Multiobjective Optimization—A Generalized Homotopy Approach*; Birkhäuser: Basel, Switzerland, 2001.
18. Loridan, P. ϵ -Solutions in Vector Minimization Problems. *J. Optim. Theory Appl.* **1984**, *42*, 265–276. [\[CrossRef\]](#)
19. Hernández, C.; Sun, J.Q.; Schütze, O. Computing the set of approximate solutions of a multi-objective optimization problem by means of cell mapping techniques. In *EVOLVE—A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation IV*; Emmerich, M., Deutz, A., Schütze, O., Bäck, T., Tantar, E., Tantar, A.A., Del Moral, P., Legrand, P., Bouvty, P., Coello Coello, C.A., Eds.; Springer: Cham, Switzerland, 2013; pp. 171–188.
20. Schütze, O.; Hernandez, C.; Talbi, E.G.; Sun, J.Q.; Naranjani, Y.; Xiong, F.R. Archivers for the Representation of the Set of Approximate Solutions for MOPs. *J. Heuristics* **2019**, *5*, 71–105. [\[CrossRef\]](#)
21. Hernández, C.I.; Schütze, O.; Sun, J.Q.; Ober-Blöbaum, S. Non-Epsilon Dominated Evolutionary Algorithm for the Set of Approximate Solutions. *Math. Comput. Appl.* **2020**, *25*, 3. [\[CrossRef\]](#)
22. Schütze, O.; Hernández, C.I. *Archiving Strategies for Multi-Objective Evolutionary Optimization Algorithms*; Springer: Cham, Switzerland, 2021.
23. Xiong, F.R.; Qin, Z.C.; Xue, Y.; Schütze, O.Q.; Ding, J.Q.S. Multi-objective optimal design of feedback controls for dynamical systems with hybrid simple cell mapping algorithm. *Commun. Nonlinear Sci. Numer. Simul.* **2014**, *19*, 1465–1473. [\[CrossRef\]](#)

24. Qin, Z.Q.; Xiong, F.R.; Hernández, C.; Fernandez, J.; Ding, Q.; Schütze, O.; Sun, J.Q. Multi-objective optimal design of sliding mode control with parallel simple cell mapping method. *J. Vib. Control* **2017**, *23*, 46–54. [\[CrossRef\]](#)
25. Peitz, S.; Dellnitz, M. A Survey of Recent Trends in Multiobjective Optimal Control–Surrogate Models, Feedback Control and Objective Reduction. *Math. Comput. Appl.* **2018**, *23*, 30. [\[CrossRef\]](#)
26. Alhato, B. Direct Power Control Optimization for Doubly Fed Induction Generator Based Wind Turbine Systems. *Math. Comput. Appl.* **2019**, *24*, 77. [\[CrossRef\]](#)
27. Torres, L.; Jiménez-Cabas, J.; Gómez-Aguilar, J.; Pérez-Alcazar, P. A Simple Spectral Observer. *Math. Comput. Appl.* **2018**, *23*, 23. [\[CrossRef\]](#)
28. Bermúdez, J.R.; López-Estrada, F.R.; Besancon, G.; Valencia-Palomo, G.; Torres, L.; Hernández, H.R. Modeling and Simulation of a Hydraulic Network for Leak Diagnosis. *Math. Comput. Appl.* **2018**, *23*, 70. [\[CrossRef\]](#)
29. Hernández, C.; Schütze, O.; Sun, J.Q. Global Multi-objective Optimization by Means of Cell Mapping Techniques. In *EVOLVE VII*; Emmerich, M., Deutz, A., Schütze, O., Legrand, P., Tantar, E., Tantar, A.A., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 25–56. [\[CrossRef\]](#)
30. Kemeny, J.; Snell, J. *Finite Markov Chains: With a New Appendix "Generalization of a Fundamental Matrix"*; Undergraduate Texts in Mathematics; Springer: Cham, Switzerland, 1976.
31. Deb, K. *Multi-Objective Optimization Using Evolutionary Algorithms*; John Wiley & Sons: Chichester, UK, 2001; ISBN 0-471-87339-X.
32. Preuss, M.; Naujoks, B.; Rudolph, G. Pareto Set and EMOA Behavior for Simple Multimodal Multiobjective Functions. In *PPSN IX*; Runarsson, T., Beyer, H.G., Burke, E., Merelo-Guervós, J.J., Whitley, L.D., Yao, X., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 513–522. [\[CrossRef\]](#)
33. Rudolph, G.; Naujoks, B.; Preuss, M. Capabilities of EMOA to Detect and Preserve Equivalent Pareto Subsets. In *EMO 2007*; Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2007; pp. 36–50. [\[CrossRef\]](#)
34. Schaeffler, S.; Schultz, R.; Weinzierl, K. Stochastic Method for the Solution of Unconstrained Vector Optimization Problems. *J. Optim. Theory Appl.* **2002**, *114*, 209–222. [\[CrossRef\]](#)
35. Schütze, O.; Esquivel, X.; Lara, A.; Coello Coello, C.A. Using the Averaged Hausdorff Distance as a Performance Measure in Evolutionary Multi-Objective Optimization. *IEEE Trans. Evol. Comput.* **2012**, *16*, 504–522. [\[CrossRef\]](#)
36. Bogoya, J.M.; Vargas, A.; Cuate, O.; Schütze, O. A (p,q)-Averaged Hausdorff Distance for Arbitrary Measurable Sets. *Math. Comput. Appl.* **2018**, *23*, 51. [\[CrossRef\]](#)
37. Bogoya, J.M.; Vargas, A.; Schütze, O. The Averaged Hausdorff Distances in Multi-Objective Optimization: A Review. *Mathematics* **2019**, *7*, 894. [\[CrossRef\]](#)
38. Fernández, J.; Schütze, O.; Hernández, C.; Sun, J.Q.; Xiong, F.R. Parallel Simple Cell Mapping for Multi-objective Optimization. *Eng. Optim.* **2016**, *48*, 1845–1868. [\[CrossRef\]](#)

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).