

# Global Multi-objective Optimization by Means of Cell Mapping Techniques

Carlos Hernández, Oliver Schütze and Jian-Qiao Sun

**Abstract** Multi-objective optimization problems (MOPs) arise in many fields in engineering. In this chapter we argue that adaptation of cell mapping techniques, originally designed for the global analysis of dynamical systems, are well-suited for the thorough analysis of low-dimensional MOPs. Algorithms of this kind deliver an approximation of the set of global solutions, the Pareto set, as well as the set of locally optimal and nearly optimal solutions in one run of the algorithm which may significantly improve the underlying decision making process. We underline the statements on some illustrative examples and present comparisons to other algorithms.

## 1 Introduction

In many applications the problem arises that several objectives have to be optimized concurrently. For instance, two important objectives in many space mission design problems are the time of flight and the cost for a mission to a certain destination [1]. One important characteristic of such a *multi-objective optimization problem* (MOP) is that its solution set, the *Pareto set*, does typically not consist of a singleton but forms a  $(k - 1)$ -dimensional object, where  $k$  is the number of objectives involved in the MOP. The computation of Pareto sets thus represents in general a challenge. Even more, in certain applications one may be interested in approximate solutions that may allow the decision maker (DM) to find alternative or backup solutions to a given problem. As for the above space mission design problem this could be a trajectory that is slightly more costly and has a slightly longer time of flight than an

---

C. Hernández (✉) · O. Schütze  
Computer Science Department, CINVESTAV-IPN, Mexico City, Mexico  
e-mail: chernandez@computacion.cs.cinvestav.mx

O. Schütze  
e-mail: schuetze@cs.cinvestav.mx

J.-Q. Sun  
School of Engineering, University of California Merced, Merced, USA  
e-mail: jsun3@ucmerced.edu

optimal one but offers in turn a different realization of the problem (see [2] for such examples where the launch date has been considered as influential for the related decision making process). The set of these approximate solutions even forms an  $n$ -dimensional set, where  $n$  is the number of decision variables involved in the model.

There exist so far many techniques for the numerical treatment of MOPs. Most of these works focus on the detection of one or several optimal solutions, while the consideration of approximate solutions is relatively scarce, probably due to the dimension of the solution set. There exists, for instance, a variety of point-wise iterative search procedures that are capable of generating a sequence of points moving either toward or along the Pareto set (e.g., [3–7] and references therein). These local methods, however, are not universally applicable due to the fact that the solution set is not a singleton as well as some possible characteristics of the model such as multi-modality and disconnectedness of the domain and/or the set of interest. A possible alternative is given by set oriented methods that are of global nature but, in turn, applicable to lower dimensional problems. Among them, specialized evolutionary strategies have caught the interest of many researchers in the recent past (e.g., [8–10]) since algorithms of this kind are very robust, applicable to a broad class of problems, and deliver a finite size approximation of the set of interest in one single run. Another class of set oriented methods are subdivision techniques [11–13] that start by considering an  $n$ -dimensional box that contains the domain of the MOP. This box gets subdivided into a set of smaller boxes, and according to certain conditions it is decided which box could contain a part of the set of interest and is thus suited for further investigation. The other, unpromising boxes, are discarded from the collection. This process, subdivision and selection, is performed on the current box collection until the desired granularity of the boxes is reached. This way, a tight covering of the Pareto set is obtained.

In this chapter, we argue that *cell mapping techniques* are in particular advantageous for the thorough investigation of low dimensional problems. Such problems occur such problems occur, for instance, in optimal control [14–17]. Cell mapping techniques were first introduced in [18] for 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. In the context of multi-objective optimization this is in particular the extended set of options that can be offered to the DM after analyzing the model. There are first of all the Pareto set and the set of approximate solutions as motivated above. In particular if there exist several possibilities to obtain the same optimal or nearly optimal performance, other methods have problems to detect them all since the notion of dominance is defined in objective space (and thus, typically only one of these solutions is detected). Further, the entire set of local optima can be identified that also serve as potential backup solutions [2] and that are interesting for landscape analysis [19]. 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 this work we will investigate adaptations of the cell mapping techniques to the context of multi-objective optimization where we will concentrate on the computation of optimal and nearly optimal solutions of a given MOP. A preliminary study of approximate solutions in the sense of Loridan [20] by means of cell mapping can be found in [21]. Further, applications of the method to the design of optimal feedback control are presented in [15, 16].

The remainder of this chapter is organized as follows: in Sect. 2, we state the notations and some background required for the understanding of the chapter. In Sect. 3, we state the cell mapping techniques for MOPs. In Sect. 4, we will present some numerical results. Finally, in Sect. 5, we conclude and will give some paths for future work.

## 2 Notations and Background

In the following we consider continuous MOPs

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

where  $Q \subset \mathbb{R}^n$  is the domain of the problem and  $F$  is defined as the vector of the objective functions  $F : Q \rightarrow \mathbb{R}^k$ ,  $F(x) = (f_1(x), \dots, f_k(x))^T$ , and where each objective  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$  is (for simplicity) sufficiently smooth.

The optimality of a MOP is defined by the concept of *dominance* [22]: a vector  $v \in \mathbb{R}^k$  is less than  $w \in \mathbb{R}^k$  ( $v <_p w$ ), if  $v_i < w_i$  for all  $i \in 1, \dots, k$ . The relation  $\leq_p$  is defined analogously. Then, a vector  $y \in Q$  is *dominated* by a vector  $x \in Q$  (in short:  $x < y$ ) with respect to (MOP) if and only if  $f_i(x) \leq f_i(y)$ ,  $i = 1, \dots, k$ , and there exists an index  $j$  such that  $f_j(x) < f_j(y)$ , else  $y$  is non-dominated by  $x$ . A point  $x \in Q$  is called (*Pareto*) *optimal* or a *Pareto point* if and only if there is no  $y \in Q$  which dominates  $x$ . The set of all Pareto optimal solutions  $P_Q$  is called the *Pareto set*, and its image  $F(P_Q)$  the *Pareto front*. Both sets typically form a  $(k - 1)$ -dimensional object.

To define the set of approximate solutions we need the following definition.

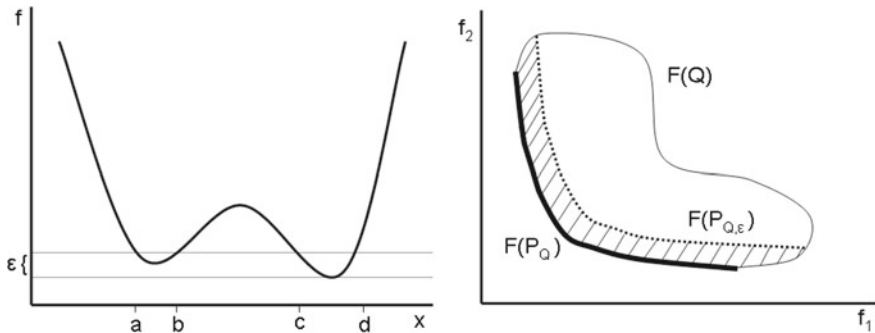
**Definition 1** ([20, 23]) Let  $\varepsilon = (\varepsilon_1, \dots, \varepsilon_k) \in \mathbb{R}_+^k$  and  $x, y \in Q$ .

- (a)  $x$  is said to  $\varepsilon$ -dominate  $y$  ( $x <_\varepsilon y$ ) with respect to (MOP) if and only if  $F(x) - \varepsilon \leq_p F(y)$  and  $F(x) - \varepsilon \neq F(y)$ .
- (b)  $x$  is said to  $-\varepsilon$ -dominate  $y$  ( $x <_{-\varepsilon} y$ ) with respect to (MOP) if and only if  $F(x) + \varepsilon \leq_p F(y)$  and  $F(x) + \varepsilon \neq F(y)$ .

The notion of  $-\varepsilon$ -dominance can be used to define our set of interest.

**Definition 2** ([23]) Denote by  $P_{Q,\varepsilon}$  the set of points in  $Q \subset \mathbb{R}^n$  that are not  $-\varepsilon$ -dominated by any other point in  $Q$ , i.e.,

$$P_{Q,\varepsilon} := \{x \in Q \mid \nexists y \in Q : y <_{-\varepsilon} x\}. \quad (1)$$



**Fig. 1** Two different examples for sets  $P_{Q, \epsilon}$ . At the *left*, we show the case for  $k = 1$  and in parameter space with  $P_{Q, \epsilon} = [a, b] \cup [c, d]$ . Note that the image solutions  $f([a, b])$  are nearly optimal (measured in objective space), but that the entire interval  $[a, b]$  is not ‘near’ to the optimal solution which is located within  $[c, d]$ . At the *right*, we show an example for  $k = 2$  in image space,  $F(P_{Q, \epsilon})$  is the approximate Pareto front (taken from [23])

The set  $P_{Q, \epsilon}$  contains all  $\epsilon$ -efficient solutions, i.e., solutions which are optimal up to a given (small) value of  $\epsilon$ . See Fig. 1 for two examples.

In [23, 24] several archiving techniques were proposed. In this work, we focus in  $ArchiveUpdateP_{Q, \epsilon}$ . The archiver guarantees convergence under certain assumptions on the MOP and the generation process (see for more details [23, 24]). Algorithm 1 show the realization of the archiver.

---

**Algorithm 1**  $A := ArchiveUpdateP_{Q, \epsilon}(P, A_0, \epsilon)$

---

**Require:** population  $P$ , archive  $A_0$

**Ensure:** updated archive  $A$

```

1:  $A := A_0$ 
2: for all  $p \in P$  do
3:   if  $\nexists a \in A : a \prec_{-\epsilon} p$  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

```

---

The cell mapping method was originally proposed by Hsu [18, 25] for global analysis of nonlinear dynamical systems in the *state space*. The cell mapping methods have been extensively studied, which lead to, the simple cell mapping, the generalized cell mapping [25], the interpolated cell mapping [26], the adjoining cell mapping [27, 28], the hybrid cell mapping [29], among others. The cell mapping methods have been applied to optimal control problems of deterministic and stochastic dynamical

systems [14, 17, 30, 31]. In [28], the cell mapping techniques were combined with dynamical systems theory in order to find all solutions to a system of nonlinear algebraic equations.

The cell mapping methods transform the point-to-point dynamics into a cell-to-cell mapping by discretizing both phase space and the integration time. The simple cell mapping (SCM) offers an effective approach to investigate global response properties of the system. The cell mapping with a finite number of cells in the computational domain will eventually lead to closed groups of cells of the period equal to the number of cells in the group. The periodic cells represent approximate invariant sets, which can be periodic motion and stable attractors of the system. The rest of the cells form the domains of attraction of the invariant sets. For more discussions on the cell mapping methods, their properties and computational algorithms, the reader is referred to the book by Hsu [25].

### 3 Global Analysis of Dynamical Systems

In this section, we first define a dynamical system, and further on the solution of a dynamical system. We also present the concept of the domain of attraction and finally we look into the simple cell mapping method that was proposed to perform a global analysis of a given dynamical system.

#### 3.1 Dynamical Systems

**Definition 3** (*Dynamical System*) A dynamical system [25] can be considered to be a model describing the temporal evolution of a system and it is defined as follows:

$$\dot{x} = G(x),$$

where  $x$  is a  $n$ -dimensional vector and  $G : \mathbb{R}^n \rightarrow \mathbb{R}^n$  is, in general, a nonlinear vector function. The evolution of such a dynamical system can be described by a function of the form:

$$x_{m+1} = G(x(m), \mu), \quad (2)$$

where  $x$  is a  $n$ -dimensional vector,  $m$  denotes the mapping step,  $\mu$  is a parameter vector, and  $G$  is a general nonlinear vector function. In this case ordinary differential equations can be used to describe the dynamical systems. These are defined as follows:

$$\dot{x} = F(x, t, \mu); \quad x \in \mathbb{R}^n, \quad t \in \mathbb{R}, \quad \mu \in \mathbb{R}^l,$$

where  $x$  is a  $n$ -dimensional state vector,  $t$  is the time variable,  $\mu$  is a  $l$ -dimensional parameter vector, and  $F$  is a vector-valued function of  $x$ ,  $t$  and  $\mu$ .

**Definition 4** (*Fixed point*) When the evolution of a dynamical system is made, one may find a point that satisfies the following:

$$x^* = G(x^*, \mu).$$

In this case,  $x^*$  is called a fixed point of Eq. (2).

**Definition 5** (*Periodic group*) A periodic solution of Eq. (2) of period  $l$  is a sequence of  $l$  distinct points  $x^*(j)$ ,  $j = 1, 2, \dots, l$  such that

$$\begin{aligned} x^*(o+1) &= G^o(x^*(1), \mu), \quad o = 1, 2, \dots, l-1, \\ x^*(1) &= G^l(x^*(1), \mu). \end{aligned} \quad (3)$$

We say that there exists a periodic solution of period  $l$ . Any of the points  $x^*(j)$ ,  $j = 1, 2, \dots, l$ , is called a periodic point of period  $l$ . One can see that a fixed point is a periodic solution with  $l = 1$ .

**Definition 6** (*Domain of attraction*) We say  $x^*(j)$  is an attractor if there exists a neighborhood  $U$  of  $x^*(j)$  such that for every open set  $V \supset x^*(j)$  there is a  $N \in \mathbb{N}$  such that  $f^j(U) \subset V$  for all  $j \geq N$ . Hence, we can restrict ourselves to the closed invariant set  $x^*(j)$ , and in this case we obtain

$$x^*(j) = \bigcap_{j \in \mathbb{N}} G^j(U).$$

Thus, we can say that all the points in  $U$  are attracted by  $x^*(j)$  (under iteration of  $G$ ), and  $U$  is called basin of attraction of  $x^*(j)$ . If  $U = \mathbb{R}^n$ , then  $x^*(j)$  is called the global attractor.

Several kinds of attractors exists, however, only the ones formed by the set of periodic solutions will be considered in this work.

### 3.2 Simple Cell Mapping

In this section, we present the simple cell mapping [25], which is useful to compute global attractors and domains of attraction of a given dynamical system.

SCM does not consider the state space to be continuous but rather as a collection of state cells, with each cell being taken as a state entity. Because of this, now we need to introduce some basic concepts regarding the new model.

**Definition 7** (*Cell state space*) A  $n$  dimensional cell space  $S$  [25] is a space whose elements are  $n$ -tuples of integers, and each element is called a cell vector or simply a cell, and is denoted by  $z$ .

The simplest way to obtain a cell structure over a given Euclidean state space is to construct a cell structure consisting of rectangular parallelepipeds of uniform size.

**Definition 8** (*Cell functions*) Let  $S$  be the cell state space for a dynamical system and let the discrete time evolution process of the system be such that each cell in a region of interest  $S_0 \subset S$  has a single image cell after one mapping step. Such an evolution process is called simple cell mapping (SCM)

$$z(n+1) = C(z(n), \mu), z \in \mathbb{Z}^N, \mu \in \mathbb{R}^l, \quad (4)$$

where  $C : \mathbb{Z}^N \times \mathbb{R}^l \rightarrow \mathbb{Z}^N$ , and  $\mu$  is a  $l$ -dimensional parameter.

**Definition 9** (*Periodic group*) A cell  $z^*$  which satisfies  $z^* = C(z^*)$  is said to be an equilibrium cell of the system. Let  $C^m$  denote the cell mapping  $C$  applied  $m$  times with  $C^0$  understood to be the identity mapping. A sequence of  $l$  distinct cells  $z^*(j), j \in l$ , which satisfies

$$z^*(m+1) = C^m(z^*(1)), m \in l-1, z^*(1) = C^l(z^*(1)), \quad (5)$$

is said to constitute a periodic group or P-Group of period  $l$  and each of its elements  $z^*(j)$  a periodic cell of period  $l$ . One can see that an equilibrium cell is a  $l = 1$  periodic group.

**Definition 10** (*Domains of attraction*) A cell  $z$  is said to be  $r$  steps away from a periodic group if  $r$  is the minimum positive integer such that  $C^r(z) = z^*(j)$ , where  $z^*(j)$  is one of the cells of that periodic group.

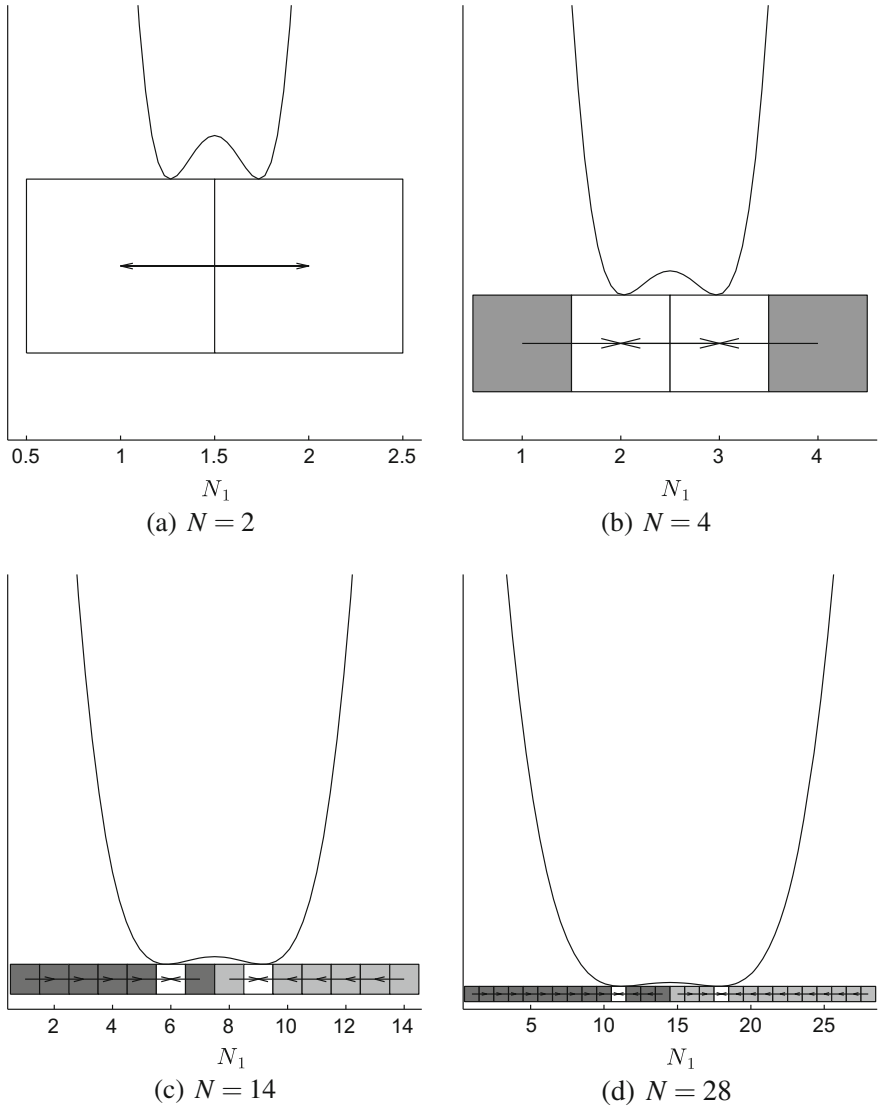
The set of all cells, which are  $r$  steps or less removed from a periodic group is called the  $r$ -step domain of attraction for that periodic group. The total domain of attraction of a periodic group is its  $r$ -step domain of attraction with  $r \rightarrow \infty$ .

The main idea of this method is based on the fact that the representation of the numbers in a computer is finite. A number does not only represent the number represented by its digits, but also an infinite neighborhood of numbers given by the precision of the machine. This does not allow to assume variables to be continuous, due to rounding errors and for this reason it is possible to consider the space as small hypercubes whose size is given by the machine precision.

The cell mapping approach [25] proposes to increase this discretization by dividing the state space into bigger hypercubes. The evolution of the dynamical system is then reduced to a new function, which is not defined in  $\mathbb{R}^n$ , but rather on the cell space. In this case we restrict ourselves to functions that are strictly deterministically defined. For this case, we have the so-called simple cell mapping method, which is effective to obtain the attractors and basins of attraction of a dynamical system.

The SCM method uses some sets in order to capture the global properties of a cell, which we describe in the following:

- Group motion number (Gr): the group number uniquely identifies a periodic motion; it is assigned to every periodic cell of that periodic motion and also to every cell in the domain of attraction. The group numbers, which are positive integers, can be assigned sequentially.



**Fig. 2** Numerical result of the SCM on Eq. (6) with different grid size, **a**  $N = 2$ ; **b**  $N = 4$ ; **c**  $N = 14$ ; **d**  $N = 28$ . The *white cells* represents the optimal solutions. The cells with the same color belong to the same domain of attraction (for those cells their mapping end in the same cell). The *arrows* represent the cell mapping. Finally, the *black curve* is the graphic representation of the problem

- **Period (Pe):** defines the period of each periodic motion.
- **Number of steps to a P-group (St):** used to indicate how many steps it takes to map this cell into a periodic cell.



According to the previous discussion, the algorithm works as follows: until all cells are processed, the value of the group motion indicates the state of the current cell and it also points out the corresponding actions to the cell.

- A value of  $Gr(cell) = 0$  means that the cell has not been processed. Hence, the state of the cell changes to “under process” and then, we follow the dynamical system to the next cell.
- A value of  $Gr(cell) = -1$  means that the cell is under process, which means we have found a periodic group and we can compute the global properties of the current periodic motion.
- A value  $Gr(cell) > 0$  means that the cell has already been processed. Hence we found a previous periodic motion along with its global properties which can be used to complete the information of the cells under process.

The cell mapping methods have been applied to optimal control problems of deterministic and stochastic dynamic systems [30–32]. Other interesting applications of the cell mapping methods include optimal space craft momentum unloading [33], single and multiple manipulators of robots [34], optimum trajectory planning in robotic systems by [35], and tracking control of the read-write head of computer hard disks [36].

Now, we present an application of the SCM on a simple example. We consider the following SOP:

$$\min_x f = 4x^3 - 2x, \quad (6)$$

where  $f : \mathbb{R} \rightarrow \mathbb{R}$  and  $x \in \mathbb{R}$ . For this problem, we have two optimal points at  $\frac{\sqrt{2}}{2}$  and  $-\frac{\sqrt{2}}{2}$ . Figure 2 shows the result for different values of  $N$  and  $Q = [-3, 3]$ . The figure shows the mapping from one cell to another one, until it reaches a periodic group. Further, it shows two different group motions and for  $N = 14$  and  $N = 28$ , we can also see where the domain of attraction of  $\frac{\sqrt{2}}{2}$  ends and the domain of attraction of  $-\frac{\sqrt{2}}{2}$  begins.

## 4 Simple Cell Mapping Techniques for MOPs

The cell mapping methods are so far designed for the global analysis of general nonlinear dynamical systems. In the following, we will present adaptations to the SCM in order to handle multi-objective optimization problems.

## 4.1 The Algorithm

First, we need an appropriate dynamical system to run SCM. For this, we propose to utilize *descent directions*. A descent direction  $v$  at a point  $x_0 \in Q$  is a direction in which all objectives improve, i.e., it holds

$$F(x_0 + tv) <_p F(x_0) \quad (7)$$

for all sufficiently small step sizes  $t \in \mathbb{R}_+$ . Such descent directions can be found e.g., in [37–40]. For the bi-objective problems presented in this chapter, we have used the following one.

**Theorem 1** ([39]) *Let (MOP) be unconstrained and be defined by two differentiable functions. If  $\nabla f_i(x_0) \neq 0$ , for  $i = 1, 2$  and for  $x_0 \in \mathbb{R}^n$ , then*

$$v(x_0) := - \left( \frac{\nabla f_1(x_0)}{\|\nabla f_1(x_0)\|} + \frac{\nabla f_2(x_0)}{\|\nabla f_2(x_0)\|} \right) \quad (8)$$

*is a descent direction at  $x_0$  of (MOP).*

Using such a descent direction, the following dynamical system

$$\dot{x}(t) = v(x(t)) \quad (9)$$

can now be used since it defines a pressure toward the Pareto set/front of the MOP at hand:  $v(x) = 0$  for every (locally) optimal point, and for all other points improvements can be found by integrating (9). Thus, the set of locally optimal Pareto points is contained in the global attractor of (9).

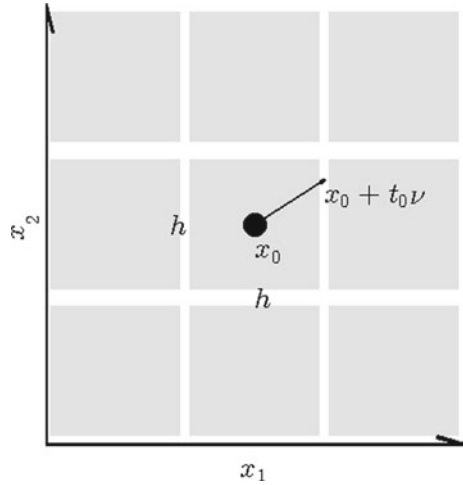
It remains to discretize the time (9), i.e., to define a ‘suitable’ step size  $t$  for the related discrete dynamical system

$$x_{i+1} = x_i + tv(x_i). \quad (10)$$

This is in general not an easy task as we have two conflicting aims. On the one hand, we would like to choose a step size  $t$  that lowers all objective functions as much as possible for a given direction  $v$ . On the other hand, it is desired to make this decision as cheap as possible in terms of computing time and number of function evaluations. One option is to use an inexact step size control as the one proposed in [38].

Here, we can take advantage of the particular setting of the multi-objective SCM. Most importantly, we have the size  $h = (ub_i - lb_i)/N_i$  for  $i = 1, \dots, n$  of the cells and know that the initial point is the center of a cell. Using this, we already have a value for sufficient decrease. If there exists a  $tv_i \geq \frac{h_i}{2}$ ,  $i = 1, \dots, n$ , then we ensure that we leave the current cell, which is required for the SCM in case the cell does not contain a part of the Pareto set. Now, to decide if the step size  $t$  is accepted, we

**Fig. 3** Illustration of the setting of the step size control problem for the SCM method



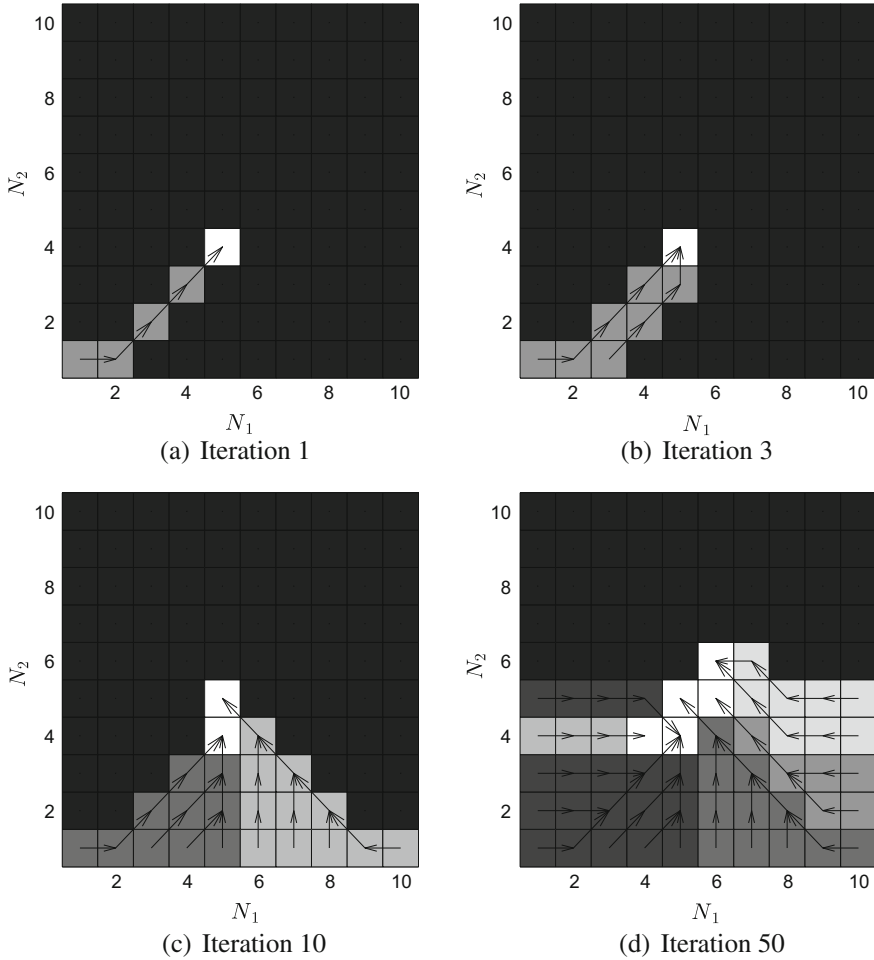
can use a dominance test. We are left with the choice of the initial step size  $t_0$ . In the current context, it is promising to compute the distance to the nearest neighbor cell given the descent direction  $\nu$  from the current cell center. Thus, we suggest to take (compare to Fig. 3):

$$t_0 = \max \left( \frac{h_i}{v_i} \right) + \varepsilon, \quad \forall i | v_i \neq 0. \tag{11}$$

We used this approach in the present work, with good computational results. Alternatively, one could use a more sophisticated method such as the one presented in [28]. The authors of this work propose an adaptive integration scheme which either finds a neighboring cell or stays in the same cell.

In both cases, although a bigger value of  $t_0$  may lead to a bigger decrease in the objective function, this value of  $t_0$  is enough to leave the current cell and we have several advantages. We would lose less information since we would be moving to a neighbor cell. Also with this step size control we would be in the frontier between the current cell and its neighbor, thus if the step size  $t_0$  is not accepted there is no need to use backtracking. Given that we would not be able to leave the current cell and also, since all cells are visited in the SCM method the advantages that bigger step sizes would have by going to an optimal solution with less function evaluations would be lost.

Inequality constraints are handled in the following (straightforward) way: if the center point  $x_i$  of cell  $i$  is violating any constraint, it will be discarded (i.e., mapped to the sink cell), else, the point will be mapped as described above. The inclusion of more sophisticated constraint handling techniques including the adequate treatment of equality constraints is the subject of ongoing study.



**Fig. 4** Iteration of the SCM. The *white cells* represent the optimal solutions found so far. The *arrows* show the path from the starting cell to an optimal solution of the MOP. The *darker cells* represent unexplored regions

Algorithm 2 shows the pseudo code of the cell mapping technique for the treatment of MOPs that contains the above discussion. Figure 4 gives some insight into the behavior of the SCM using the MOP CONV2 (see Table 1) on a  $10 \times 10$  grid. The figure shows the result of the SCM after 1, 3, 10 and 50 iterations in cell space. First, we look at the cell located in (1, 1), which has been taken as the starting cell. Next, we can follow the mapping from this cell by following its arrow. These arrows are formed as follows: We take the center point of the current cell, then we apply the dynamical system (e.g., the descent direction method that we have chosen) on the center point and finally, with the new solution found, we compute to which cell it

**Table 1** MOPs considered in this work

<p>WITTING</p>	<p><math>F(x) = (f_1(x), f_2(x))</math>, where:</p> $f_1(x) = \frac{1}{2}(\sqrt{1 + (x + y)^2} + \sqrt{1 + (x - y)^2} + x - y) + \lambda \cdot e^{-(x-y)^2}$ $f_2(x) = \frac{1}{2}(\sqrt{1 + (x + y)^2} + \sqrt{1 + (x - y)^2} - x + y) + \lambda \cdot e^{-(x-y)^2}$	<p><math>-10 \leq x_1 \leq 10</math>  <math>-10 \leq x_2 \leq 10</math></p>
<p>CONV2</p>	<p><math>F(x) = (f_1(x), f_2(x))</math>, where:</p> $f_1(x) = (x_1 - 1)^4 + (x_2 - 1)^2$ $f_2(x) = (x_1 + 1)^2 + (x_2 + 1)^2$	<p><math>-3 \leq x_1 \leq 3</math>  <math>-3 \leq x_2 \leq 3</math></p>
<p>CONV3</p>	<p><math>F(x) = (f_1(x), f_2(x))</math>, where:</p> $f_1(x) = (x_1 - 1)^4 + (x_2 - 1)^2 + (x_3 - 1)^2$ $f_2(x) = (x_1 + 1)^2 + (x_2 + 1)^4 + (x_3 + 1)^2$ $f_3(x) = (x_1 - 1)^2 + (x_2 + 1)^4 + (x_3 - 1)^4$	<p><math>-3 \leq x_1 \leq 3</math>  <math>-3 \leq x_2 \leq 3</math>  <math>-3 \leq x_3 \leq 3</math></p>
<p>RUDOLPH</p>	<p><math>F(x) = (f_1(x), f_2(x))</math>, where:</p> $f_1(x) = (x_1 - t_1(c + 2a) + a)^2 + (x_2 - t_2b)^2 + \delta_t$ $f_2(x) = (x_1 - t_1(c + 2a) - a)^2 + (x_2 - t_2b)^2 + \delta_t$ <p>where</p> $t_1 = \text{sgn}(x_1) \min \left( \left\lceil \frac{ x_1  - a - c/2}{2a + c} \right\rceil, 1 \right),$ $t_2 = \text{sgn}(x_2) \min \left( \left\lceil \frac{ x_2  - b/2}{b} \right\rceil, 1 \right),$ $\delta_t = \begin{cases} 0 & \text{for } t_1 = 0 \text{ and } t_2 = 0 \\ 0.01 & \text{for } t_1 = -1 \text{ and } t_2 = 0 \\ 0.02 & \text{for } t_1 = 1 \text{ and } t_2 = 0 \\ 0.03 & \text{for } t_1 = 0 \text{ and } t_2 = -1 \\ 0.04 & \text{for } t_1 = -1 \text{ and } t_2 = -1 \\ 0.05 & \text{for } t_1 = 1 \text{ and } t_2 = -1 \\ 0.06 & \text{for } t_1 = 0 \text{ and } t_2 = 1 \\ 0.07 & \text{for } t_1 = -1 \text{ and } t_2 = 1 \\ 0.08 & \text{for } t_1 = 1 \text{ and } t_2 = 1 \end{cases}$	<p><math>a = 0.5</math>  <math>b = 5</math>  <math>c = 5</math>  <math>-10 \leq x_1 \leq 10</math>  <math>-10 \leq x_2 \leq 10</math></p>
<p>SSW</p>	<p><math>F(x) = (f_1(x), f_2(x))</math>, where:</p> $f_1(x) = \sum_{j=1}^n x_j,$ $f_2(x) = 1 - \prod_{j=1}^n (1 - w_j(x_j)),$ $w_j(z) = \begin{cases} 0.01 \cdot \exp(-(\frac{z}{20})^{2.5}) & \text{for } j = 1, 2 \\ 0.01 \cdot \exp(-\frac{z}{15}) & \text{for } j > 2 \end{cases}$	<p><math>0 \leq x_1 \leq 40</math>  <math>0 \leq x_2 \leq 40</math>  <math>0 \leq x_3 \leq 40</math></p>
<p>TANAKA</p>	<p><math>F(x) = (f_1(x), f_2(x))</math>, where:</p> $f_1(x) = x_1$ $f_2(x) = x_2$	<p><math>0 \leq x_1 \leq \pi</math>  <math>0 \leq x_2 \leq \pi</math>  <math>g_1(x) = -x_1^2 - x_2^2 + 1</math>  <math>+ 0.1 \cos(16 \tan(\frac{x_1}{x_2}))</math>  <math>g_2(x) = (x_1 - \frac{1}{2})^2</math>  <math>+ (x_2 - \frac{1}{2})^2 - 1/2</math></p>

belongs. In our example, the path is formed by the cells (1, 1), (2, 1), (3, 2), (4, 3), and (5, 4). Cell (5, 4) is an endpoint in this case, since there is not an arrow from this cell to another cell, which means we have a periodic group of 1. All the cells processed belong to the same domain of attraction and, therefore, they should have the same group number. Since, this is the first group motion discovered, we assign to it the group number 2 (the group number 1 is reserved for those periodic motions that go to the sink cell). Once we have the global properties of those cells, we have to choose a new starting cell. Since the cell (2, 1) has already been processed, we skip it and continue with the cell (3, 1). The mapping of this cell also finishes in the cell (5, 4). Thus, this cell together with the new path should have the same group number as before (group number 2).

Then, we choose a new starting cell and continue until we finish processing all the cells. As we process the cells, we gather more information of the problem. For this example we have 8 periodic motions with the same number of optimal solutions.

After one run of the SCM the information of the sets of interest can be extracted. In the following, we will do this for optimal and nearly optimal solutions.

## 4.2 Computing the Pareto Set

Since the Pareto set of a MOP is contained in the global attractor of the dynamical system that is derived from a descent direction, all cells with periodic groups are at first point interesting. That is, such cells can potentially contain a part of the Pareto set. It is important to note that due to the properties of the dynamical system periodic groups with size larger than 1 should not appear, however, due to the discretization both in space and time exactly this happens (i.e., oscillations around Pareto optimal solutions can be observed leading to such periodic groups). Hence, we also consider those cells as candidates. The collection of those cells form the candidate set.

This collection can then be further investigated (e.g., via a more fine grain cell mapping or via subdivision techniques), or an approximation of the Pareto set can be directly determined via the center points of the boxes (e.g., via a non-dominance test). Technically speaking, we introduce a set called *cPs* (see Algorithm 2). Candidate optimal cells are thus those cells with  $St = 0$  and  $Gr \neq 1$ .  $St = 0$  means they are part of a periodic group and  $Gr \neq 1$  ensures we do not add cells that map to the sink cell.

**Algorithm 2** Simple cell mapping for MOPs

**Require:** MOP  $F$ , Dynamical system  $v$ , upper bound  $ub$ , lower bound  $lb$ , divisions per dimension  $N$ , cell size  $h = (ub_i - lb_i)/N_i$  for  $i = 1, \dots, n$ , Total number of cells  $N_c = N_1 \times N_2 \times \dots \times N_n$  for  $i = 1, \dots, n$

**Ensure:** Set of cells  $z$ , image of cells  $C$ , group number  $Gr$ , period number  $Pe$ , step number  $St$ , candidate pareto set  $cPs$

```

1:  $current\_group \leftarrow 1$ 
2:  $cPs = \{\}$ 
3:  $Gr(i) \leftarrow 0, \forall i \in N_c$ 
4: for all  $pcell \in N_c$  do
5:    $cell \leftarrow pcell$ 
6:    $i \leftarrow 0$ 
7:   while  $newcell = true$  do
8:      $x_i \leftarrow$  center point of  $cell$ 
9:     if  $Gr(cell) = 0$  then
10:       $v \leftarrow$  compute as in Eq. (9)
11:       $t \leftarrow$  compute as in Eq. (11)
12:       $p_{i+1} \leftarrow x_i + vt$ 
13:       $ncell \leftarrow$  cell where  $p_{i+1}$  is located
14:       $C(cell) \leftarrow ncell$ 
15:       $cell \leftarrow ncell$ 
16:       $i \leftarrow i + 1$ 
17:     end if
18:     if  $Gr(cell) > 0$  then
19:        $Gr(C^j(pcell)) \leftarrow Gr(cell), j \leftarrow 0, \dots, i$ 
20:        $Pe(C^j(pcell)) \leftarrow Pe(cell), j \leftarrow 0, \dots, i$ 
21:        $St(C^j(pcell)) \leftarrow St(cell) + i - j, j \leftarrow 0, \dots, i$ 
22:        $cell \leftarrow C(cell)$ 
23:        $newcell \leftarrow false$ 
24:     end if
25:     if  $Gr(cell) = -1$  then
26:        $current\_group \leftarrow current\_group + 1$ 
27:        $Gr(C^k(pcell)) \leftarrow current\_group, k \leftarrow 0, \dots, i$ 
28:        $j \leftarrow$   $i^{th}$  value when period appears
29:        $Pe(C^k(pcell)) \leftarrow i - j, k \leftarrow 0, \dots, i$ 
30:        $St(C^k(pcell)) \leftarrow j - k, k \leftarrow 0, \dots, j - 1$ 
31:        $St(C^k(pcell)) \leftarrow 0, k \leftarrow j, \dots, i$ 
32:        $cPs \leftarrow cPs \cup cell_k, k \leftarrow j, \dots, i$ 
33:        $cell \leftarrow C(cell)$ 
34:        $newcell \leftarrow false$ 
35:     end if
36:   end while
37: end for

```

### 4.3 Computing the Set of Approximate Solutions

Once the run of the SCM is performed, also the approximate solutions can be detected by analyzing the given data. For instance, if an approximation of  $P_{Q,\varepsilon}$  is desired, one can use the archiving technique *ArchiveUpdate* $P_{Q,\varepsilon}$  presented in [23, 24]. In order to prevent that all points have to be considered in the archive, one can proceed as follows:

Once the group number of the current periodic motion is discovered, the idea is to update the archive first with the periodic group of the current periodic motion and to continue with the rest of the periodic motion. Once it finds a cell which is not in  $P_{Q,\varepsilon}$  it stops the procedure. This can be done since subsequent points are dominated by the current candidate solution and thus also not a member of  $P_{Q,\varepsilon}$ . In the worst case, however, this algorithm visits all the cells and the archive has to be updated by all candidate solutions. This is the case if the entire domain  $Q$  is equal to  $P_{Q,\varepsilon}$ . Typically, this is not the case and the number of center points considered by the archive is much lower than the total number of cells.

### 4.4 Error Estimates

Since SCM evaluates the entire discretized search space in one run of the algorithm, we are able to provide estimations on the maximal error that can occur in the approximation of the set of interest. Since we are particularly interested in errors of the Pareto front (i.e., errors in objective space) the following estimates are based on Lipschitz continuity.

Assume in the following that the objective function  $F$  is Lipschitz continuous on each cell, i.e.,

$$\|F(x) - F(y)\| \leq L_{B(c,r)} \|x - y\|, \quad \forall x, y \in B(c, r), \quad (12)$$

where

$$B(c, r) := \{y \in \mathbb{R}^n : |c_i - y_i| \leq r_i, i = 1, \dots, n\} \quad (13)$$

is a cell (or generalized box) with center  $c$  and radius  $r$ , and  $L_{B(c,r)}$  is the Lipschitz constant of  $F$  within  $B(c, r)$ . Since SCM evaluates cells at the center  $c$  and since the maximal distance on the right hand side of (12) is given for vertices of the cell, e.g.,  $y = c + r$ , we can estimate (12) at least for unconstrained problems by

$$\|F(c) - F(y)\| \leq L_{B(c,r)} \|r\|, \quad \forall y \in B(c, r). \quad (14)$$

The above formula might already be used to measure errors in image space. In the context of multi-objective optimization, however, a potential trouble is that some objectives may be in completely different ranges (e.g., the model SSW that will be



considered in the next section. See Fig. 4 for an approximation of the Pareto front). We suggest hence to consider the error bounds for each objective, that is

$$\|F_i(c) - F_i(y)\| \leq L_{B(c,r)}^{(i)} \|r\|, \quad \forall y \in B(c, r), \quad i = 1, \dots, k, \quad (15)$$

where  $L_{B(c,r)}^{(i)}$  is the Lipschitz constant for objective  $f_i$ . If the boxes are small enough, one may approximate this value by the absolute value of the gradient at the center point leading to the estimate

$$E(B(c, r), f_i) := |\nabla f_i(c)| \|r\|, \quad \forall y \in B(c, r), \quad i = 1, \dots, k, \quad (16)$$

which we use in this study. As errors for the entire approximation we thus define

$$E_i := \max_{E(c,r) \in Q} E(B(c, r), f_i), \quad i = 1, \dots, k. \quad (17)$$

It remains to measure the approximation quality obtained via SCM on a particular problem after the algorithm has been performed. For this, we think it makes sense to measure the distance of the Pareto front to the image  $F(A)$  of the archive  $A$  of candidate solutions (e.g., the nondominated solutions) obtained by SCM. The *Inverted Generational Distance* (IGD, see [41]) is widely used as a performance indicator in multi-objective optimization, and measures the (averaged) distance of the Pareto front to  $F(A)$  as

$$IGD_p(F(A), PF) = \left( \frac{1}{M} \sum_{j=1}^M \text{dist}(F_j, F(A))^p \right)^{1/p}, \quad (18)$$

where  $PF = \{F_1, \dots, F_M\}$  is a discretization of the Pareto front,  $F(A) = \{y_1, \dots, y_N\}$ ,  $\text{dist}(a, B) = \min_{b \in B} \|b - a\|$  the distance from point  $a$  to set  $B$ , and  $p \in \mathbb{N}$ . To obtain an error bound for the objective space of each objective function  $f_i$ , one can modify IGD as follows:

$$IGD_p^{(i)}(F(A), PF) = \left( \frac{1}{M} \sum_{j=1}^M \text{dist}(F_{j,i}, F(A)_i)^p \right)^{1/p}, \quad i = 1, \dots, k, \quad (19)$$

where  $F_{j,i}$  denotes the  $i$ -th entry of  $F_j$ , and  $F(A)_i = \{y_{1,i}, \dots, y_{N,i}\}$ . Note that a finite value of  $p$  in (19) averages the distances from  $F_j$  to  $F(A)$ . If this is not wanted, one can choose  $p = \infty$  leading to

$$IGD_\infty^{(i)}(F(A), PF) = \max_{i=1, \dots, M} \text{dist}(F_{j,i}, F(A)_i), \quad i = 1, \dots, k. \quad (20)$$

In this study, we will consider  $p = 1$ .

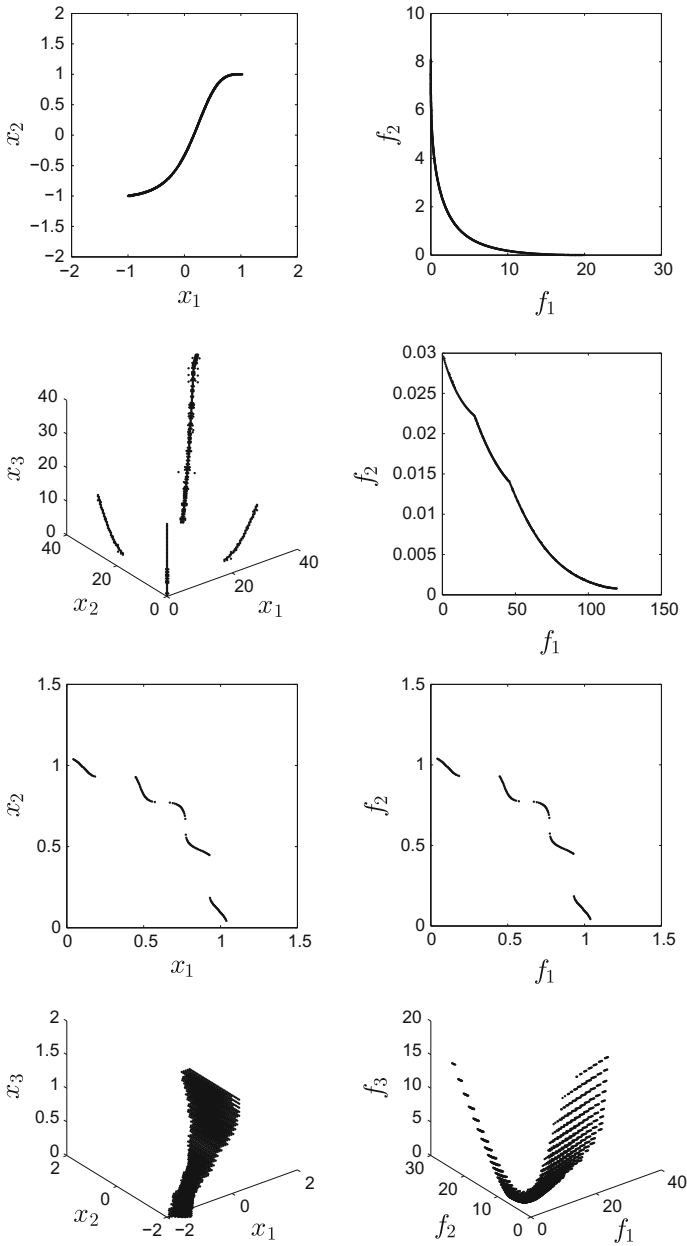
## 5 Numerical Results

In the following we present some results obtained by the cell mapping techniques and make some comparisons to other techniques.

First we consider the capability of the SCM to compute approximations of the global Pareto set. Figure 5 shows some results obtained by SCM on four MOPs (see Table 1 for the description of the problems). CONV2 is a convex bi-objective problem. The Pareto set is a curve connecting the points  $(-1, -1)^T$  and  $(1, 1)^T$ . SSW is a bi-objective problem 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. TANAKA is a bi-objective inequality constrained problem whose Pareto front is disconnected and has convex and concave parts. Finally, CONV3 is a convex tri-objective problem. We have used a  $1,000 \times 1,000$  grid to perform the cell mapping for the problems with  $n = 2$  and a  $100 \times 100 \times 100$  grid for CONV3. Figure 4 shows the results. In all cases SCM is able to obtain a fine grained approximation of both Pareto set and front. Table 2 shows the error estimate discussed in the previous section on six MOPs including the four ones considered in Fig. 5 on different grid sizes together with their IGD values. The match of both values (for each objective value) is almost perfect for WITTING while the IGD values for the other unconstrained problems are (much) better than the  $E_i$  values since these describe the worst case scenario. An exception is the TANAKA problem where the IGD values are worse. The reason is that this problem is constrained, and in this case the estimation made in (14) does not have to hold: it may happen that a cell contains a part of the Pareto set, but its center point is not feasible and will thus be discarded by SCM. Thus, the IGD values can get larger than the estimation made in (14) which is the case for TANAKA.

Tables 3 and 4 show comparisons of the SCM with NSGA-II [42] and MOEA/D [43], two state-of-the-art MOEAs. For the comparison we have used a budget of 60,000 function evaluations for all algorithms, and to measure the approximation quality we have used the averaged Hausdorff distance  $\Delta_1$  [44] which is more common for the comparison of different algorithms (in particular, since it gives *one* value for each problem). As anticipated, SCM cannot outperform the MOEAs (which can get similar approximation qualities even for much smaller budgets of function evaluations). Nevertheless, SCM is competing in the approximation of the Pareto *set*.

Now, we present a comparison of the SCM with an enumeration algorithm that generates all solutions with a certain precision and then we keep the nondominated solutions. The comparisons were made on CONV2 and RUDOLPH. In order to do this comparison, we use a refinement method on SCM. This can be done due to the fact that SCM returns a set of boxes and then we can use this set and apply subdivision techniques.



**Fig. 5** Pareto sets (*left*) and fronts (*right*) obtained by the multi-objective SCM on the problems CONV2, SSW, TANAKA, and CONV3 (from *above* to *below*)

**Table 2** Error of a given cell measured on maximum error ( $k_i$  err), error on the optimal points ( $k_i$  err opt) and  $IGD_i$

Problem	Size of grid	k1 err	k2 err	k3 err	k1 err opt	k2 err opt	k3 err opt	$IGD^1$	$IGD^2$	$IGD^3$
WITTING [45]	100 × 100	0.223	0.223		0.1999	0.1999		0.1984	0.1984	
	200 × 200	0.1115	0.1115		0.0999	0.0999		0.0991	0.0991	
	500 × 500	0.0446	0.0446		0.04	0.04		0.0396	0.0396	
CONV2 [46]	1000 × 1000	0.0223	0.0223		0.02	0.02				
	100 × 100	10.624	0.4764		1.348	0.2424		0.8288	0.1683	
	200 × 200	5.3724	0.2391		0.6892	0.124		0.4075	0.0485	
CONV3 [46]	500 × 500	2.1635	0.0959		0.2745	0.0492		0.1081	0.0725	
	1000 × 1000	1.0842	0.048		0.1366	0.0241				
	20 × 20 × 20	105.4206	105.4206	148.9384	32.6077	14.6249	14.4099	2.1294	3.548	1.981
RUDOLPH [47]	30 × 30 × 30	76.124	76.124	107.5589	24.4209	8.3636	14.8069	2.0873	3.3505	2.0557
	60 × 60 × 60	41.1418	41.1418	58.1362	10.8723	4.873	6.4499	1.5386	1.5386	1.3699
	100 × 100 × 100	25.4513	25.4513	35.9657	6.5238	3.4837	3.4519			
SSW [48]	100 × 100	1.8627	1.8627		0.2843	0.2843		0.1264	0.1264	
	200 × 200	0.9413	0.9413		0.1345	0.1345		0.0494	0.0494	
	500 × 500	0.3789	0.3789		0.0566	0.0566		0.0216	0.0216	
TANAKA [49]	1000 × 1000	0.1899	0.1899		0.028	0.028				
	20 × 20 × 20	6	0.0031		6	0.0031		2.4	0.0005	
	30 × 30 × 30	4	0.0021		4	0.002		1.4	0.0004	
CONV3 [46]	60 × 60 × 60	2	0.0011		2	0.001		0.4	0.0002	
	100 × 100 × 100	1.2	0.0007		1.2	0.0006				
	100 × 100	0.0222	0.0222		0.0222	0.0222		0.055	0.055	
CONV2 [46]	200 × 200	0.0111	0.0111		0.0111	0.0111		0.044	0.044	
	500 × 500	0.0044	0.0044		0.0044	0.0044		0.011	0.011	
	1000 × 1000	0.0022	0.0022		0.0022	0.0022				

**Table 3**  $\Delta_1$  values for the distances of the images of the candidate sets to  $P_Q$ 

MOP	SCM	NSGA-II	MOEA/D
WITTING	<b>0.1421</b>	0.2925	3.9561
CONV2	0.1329	<b>0.1297</b>	0.1677
CONV3	<b>0.2814</b>	1.2607	1.7306
RUDOLPH	0.1414	<b>0.0232</b>	0.4970
SSW	<b>2.6533</b>	3.7666	32.8963

**Table 4**  $\Delta_1$  values for the distances of the images of the candidate sets to  $F(P_Q)$ 

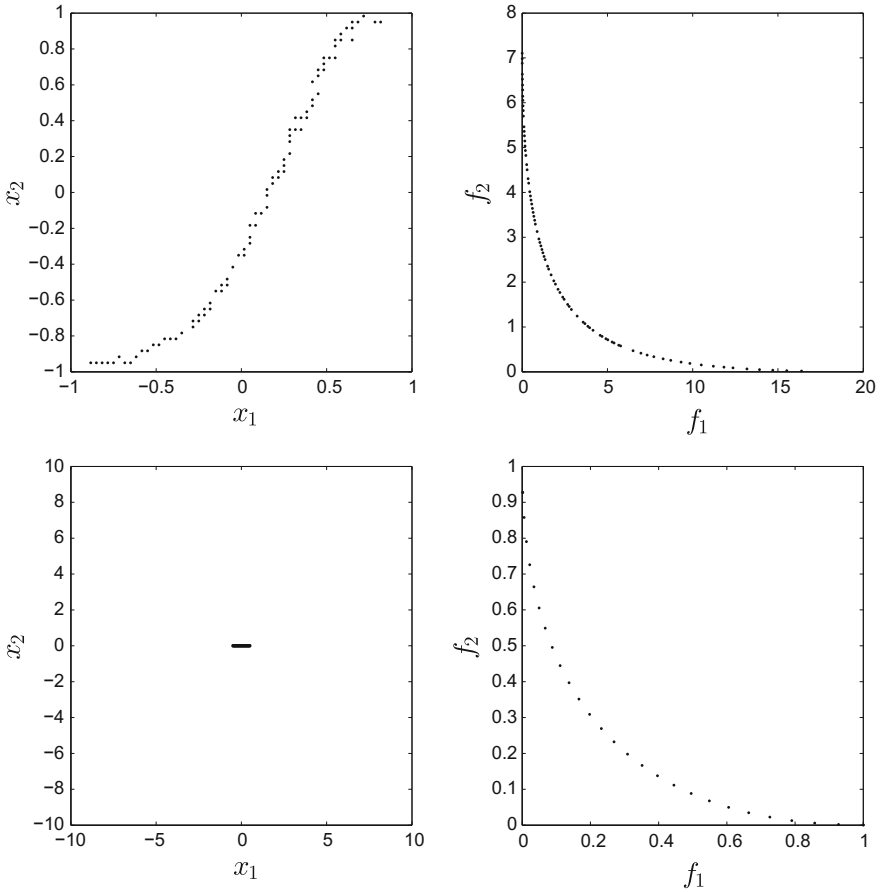
MOP	SCM	NSGA-II	MOEA/D
WITTING	<b>0.1984</b>	0.2779	5.5826
CONV2	0.8288	<b>0.1578</b>	2.5493
CONV3	<b>5.5185</b>	16.3426	27.1944
RUDOLPH	0.1649	<b>0.0282</b>	0.7863
SSW	2.4000	<b>0.8969</b>	54.0516

For CONV2 problem, we used a  $20 \times 20$  grid, and 3 subdivision steps using a  $2 \times 2$  grid of test points to evaluate each cell for SCM, which leads to 7,416 function evaluations. In the case of the grid search, we used a  $92 \times 92$  grid leading to 8,464 function evaluations. The results on terms of  $\Delta_1$  for parameter space and objective space are as follows: SCM, 0.1289 and 0.5666; grid search, 0.1529 and 1.3074 respectively. Since for  $\Delta_1$  smaller numbers represent better approximations, we can say that SCM outperforms the grid search in this example.

For RUDOLPH problem, we used a  $20 \times 20$  grid, and 3 subdivision steps using a  $3 \times 3$  grid of test points to evaluate each cell in the case of the SCM, which leads to 4,128 function evaluations. In the case of the grid search, we used a  $65 \times 65$  grid leading to 4,225 function evaluations. The results in terms of  $\Delta_1$  for parameter space and objective space are as follows: SCM, 0.0414 and 0.0724; grid search, 5.9615 and 0.1246, respectively.

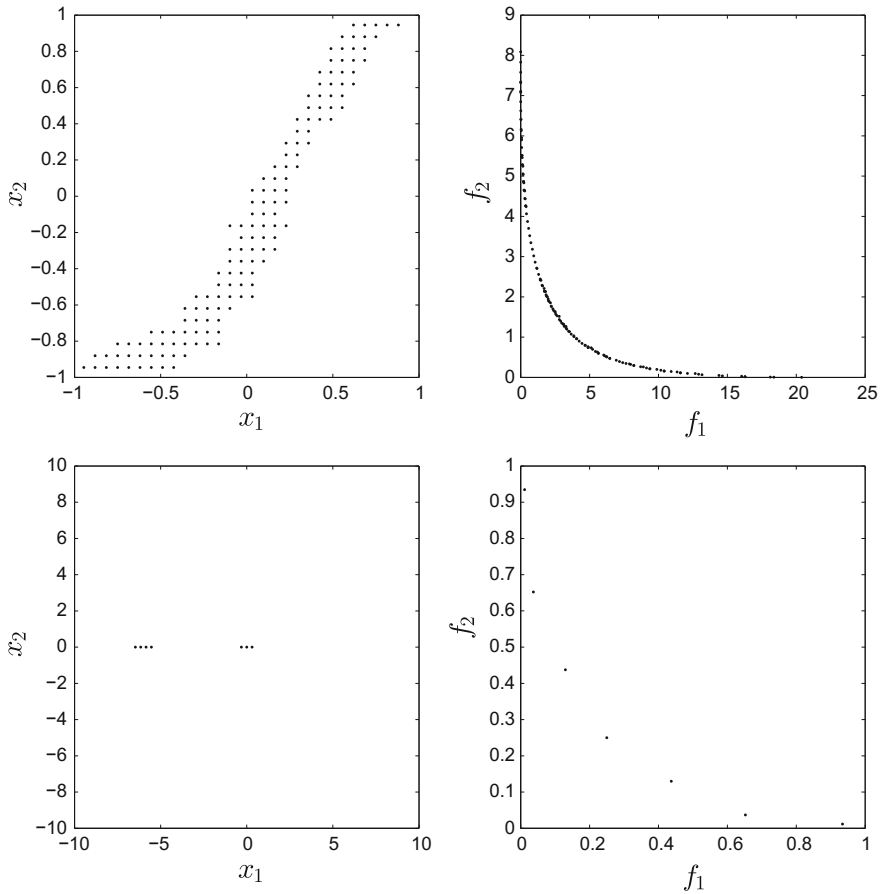
Figure 6 shows the results of the SCM method with subdivision techniques and Fig. 7 shows the results of the enumeration algorithm. In this case SCM shows a better performance in both problems, which is underlined by the indicator values. Also the SCM has advantages if more than ‘just’ the solution set is sought as the following examples demonstrate.

In some applications, it is desired to have a technique capable of computing both global and local Pareto solutions. This can be useful in cases where the global criteria does not account for all decision makers expectations [50–52].



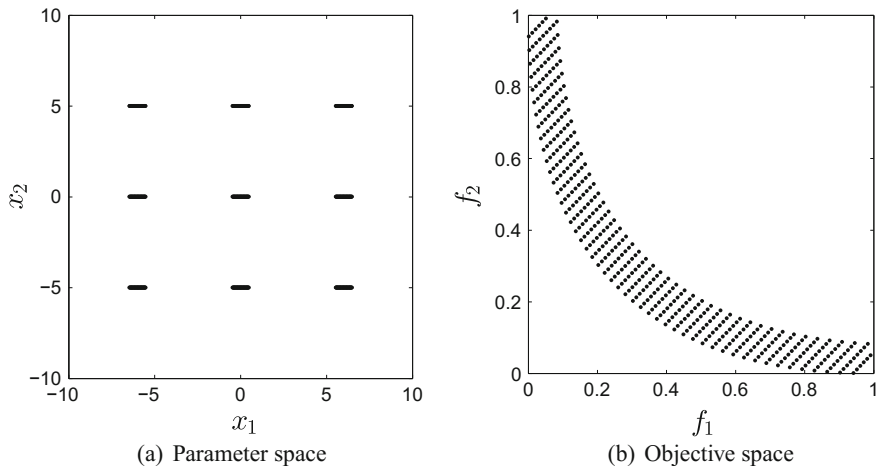
**Fig. 6** Pareto sets (*left*) and fronts (*right*) using SCM with subdivision. *Above*, we can see the results of CONJ2 and *below* the ones of RUDOLPH

One interesting example in this context is MOP RUDOLPH (see Table 1). In its original version proposed in [47] the Pareto set consists of nine different connected components, each of them mapping to the same Pareto front. We have modified this problem here slightly so that the Pareto set consists of one of these connected components whereas the other eight components map to a slightly higher value (more precisely, the objective values are shifted by a multiple of 0.01). Since this change in objective space is just slight, all nine components are hence potentially interesting for the decision maker if he/she is willing to accept this deterioration. Figure 8 shows the result of the SCM on this problem on a  $1,000 \times 1,000$  grid. Note that the algorithm is capable of detecting all nine connected components, and that each component is approximated with the same quality.



**Fig. 7** Pareto sets (*left*) and fronts (*right*) using grid search on the problems CONV2 (*above*) and RUDOLPH (*below*)

Following this idea of solving multimodal problems, we now compare the SCM method with a simple multi start algorithm that uses the same descent direction than SCM. We used a  $20 \times 20$  grid, a subdivision of  $3 \times 3$  for each optimal cell and with 3 levels of subdivision in the case of the SCM, which leads to 12,120 function evaluations. For the multi start algorithm, we used 130 starting points leading to 12,785 function evaluations. Figure 9 shows the results of the SCM and the multi start algorithm on MOP RUDOLPH with a budget of 12,500 function evaluations. The results show that SCM is able to compute evenly spread solutions while the result of the multi start approach reveals some gaps in the fronts. The results in terms of  $\Delta_1$  for parameter space and objective space are as follows: SCM, 0.1775 and 0.3362; multi start, 0.4112 and 0.5631, respectively.

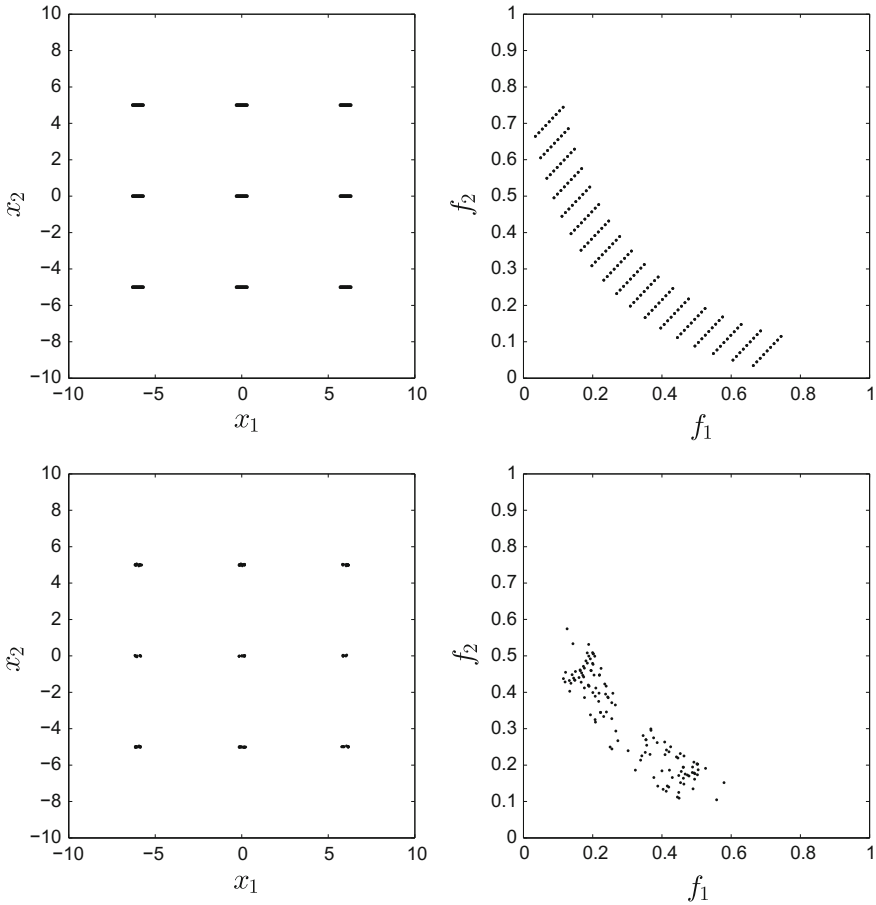


**Fig. 8** Numerical result of the SCM for MOP RUDOLPH on a  $1,000 \times 1,000$  grid

Next we address the problem to approximate the set of nearly optimal solutions. Figure 10 shows the result of the SCM on four MOPs resulting from a consideration of a  $1,000 \times 1,000$  grid. As mentioned above, the investigation of the entire set of approximate solutions is scarce. So far, archiving techniques exist that aim for the approximation of  $P_{Q,\varepsilon}$  [24], but efficient algorithms for their computations are still missing. We stress that so far many heuristics exist that utilize the concept of  $\varepsilon$ -dominance (e.g., [53–57]), however, all of them use this concept to obtain a finite size approximation of the Pareto front and *not* to obtain the set of approximate solutions. That is these works use it as a mean to improve diversity and thus get better a better approximation of the Pareto front.

In order to obtain a comparison, we have coupled NSGA-II and MOEA/D with the archiver *ArchiveUpdate* $P_{Q,\varepsilon}$  [24]. The coupling can thus be viewed as an algorithm for the computation of  $P_{Q,\varepsilon}$ , but since both MOEAs are elitist algorithms their search naturally focuses on  $P_Q$  and not on the nearly optimal solutions. Further improvements of the evolutionary strategies can thus be obtained via further modifications of the selection operators which are, however, neither straightforward nor in the scope of this chapter. Figure 11 shows numerical results obtained by the NSGA-II variant on the same MOPs, and Tables 5 and 6 show averaged  $\Delta_1$  values of the approximation qualities in parameter and objective space, respectively. For the latter we have chosen a budget of 10,000 function calls for each algorithm. As it can be seen, SCM offers the best performance in particular for the approximation of the set of interest in decision space (which is of great interest for the decision maker as motivated in Sect. 1).

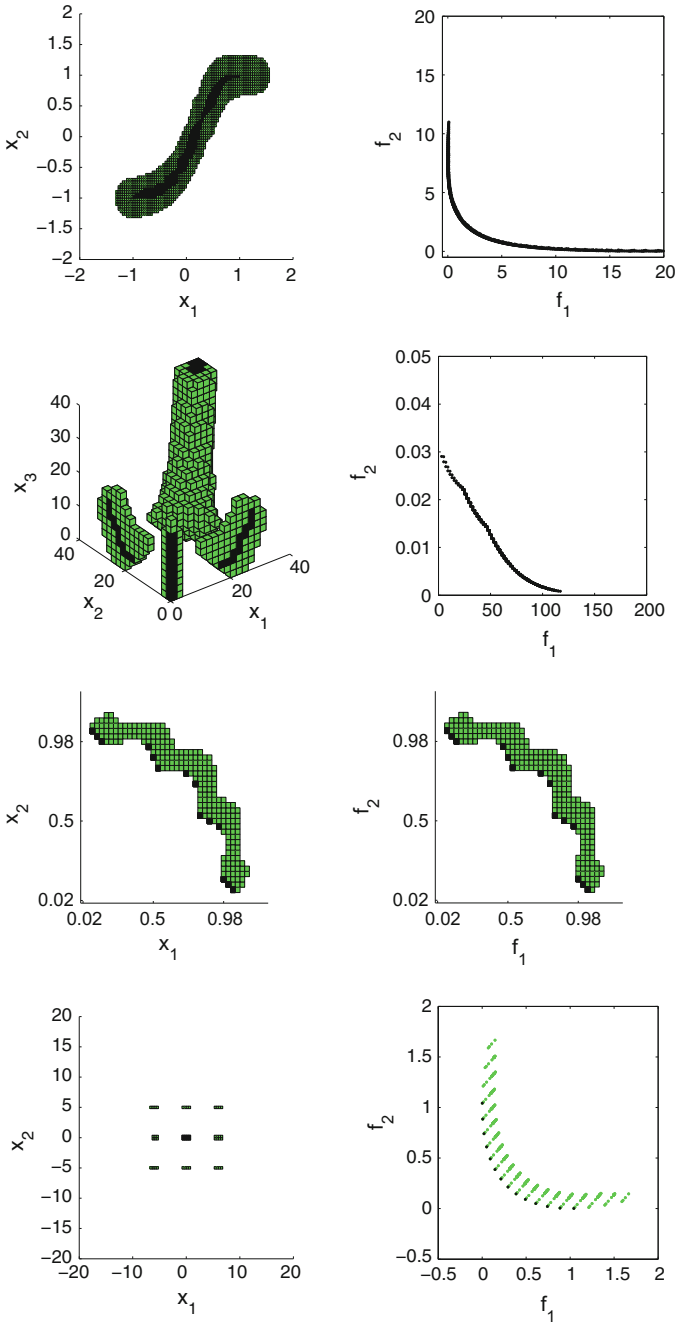




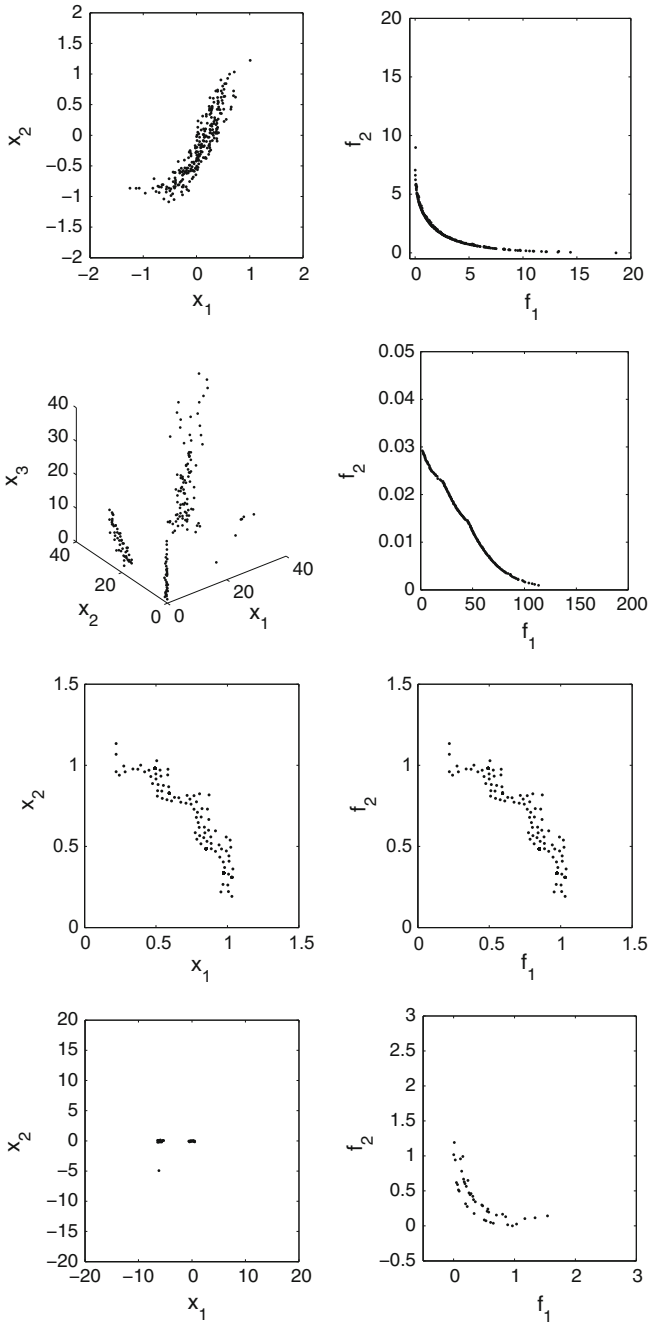
**Fig. 9** Comparison of the SCM method (*above*) and multi start (*below*) on RUDOLPH

We would like to stress that  $P_{Q,\varepsilon}$  is one way to define approximate solutions, but there exist other ways depending on the given situation, and that the data obtained by SCM is sufficient to comply with all of them. Figure 12 shows the approximate solutions obtained by SCM for different sets of approximate solutions. Next to  $P_{Q,\varepsilon}$  we have selected the notion of Tanaka [58] and Bonnel [59].

As a hypothetical decision making problem we reconsider MOP RUDOLPH. Assume for this purpose that the DM is interested in the performance  $Z = [0.17, 0.37]^T$  (measured in objective space) and further that he/she is willing to accept a deterioration of  $\varepsilon = [0.1, 0.1]$ . Then, for instance the representatives of the cells whose images are within the target regions can be presented to the DM leading here to 23 candidate solutions (i.e., the ‘optimal’ one plus another 22 nearly optimal ones) that are shown in Fig. 13. The solutions are well-spread and come in



**Fig. 10** Approximations of  $P_{Q,\varepsilon}$  (left) and  $F(P_{Q,\varepsilon})$  (right) obtained by the multi-objective SCM on the problems CONV2, SSW, TANAKA, and RUDOLPH (from above to below). In black the cells that contain Pareto optimal solutions, and in green the nearly optimal ones that do not contain a part of the Pareto set/front



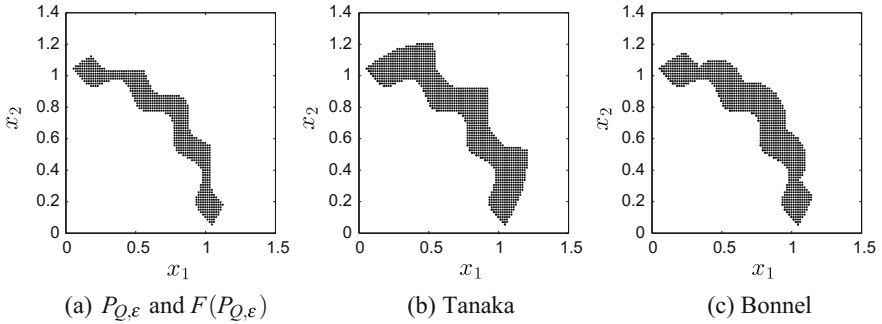
**Fig. 11** Approximations of  $P_{Q,\epsilon}$  (left) and  $F(P_{Q,\epsilon})$  (right) obtained by NSGA-II coupled with  $ArchiveUpdateP_{Q,\epsilon}$ . The problems are the same as in Fig. 10

**Table 5**  $\Delta_1$  values for the distances of the candidate solution set to  $P_{Q,\epsilon}$ , the best solutions in boldface

MOP	MOEA/D	NSGA-II	SCM
CONV2	0.5141	0.4628	<b>0.0849</b>
RUDOLPH	7.2438	8.0552	<b>0.2102</b>
SSW	10.8365	10.9384	<b>0.8660</b>
TANAKA	0.1462	0.1371	<b>0.0248</b>

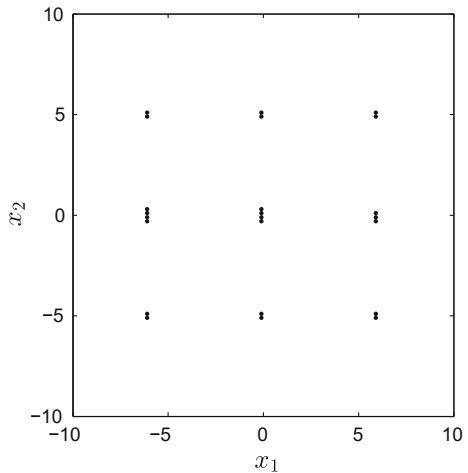
**Table 6**  $\Delta_1$  values for the distances of the images of the candidate sets to  $F(P_{Q,\epsilon})$ , the best solutions in boldface

MOP	MOEA/D	NSGA-II	SCM
CONV2	7.8902	8.0027	<b>2.4250</b>
RUDOLPH	0.5090	0.7390	<b>0.2186</b>
SSW	5.8152	2.6852	<b>1.5000</b>
TANAKA	0.1462	0.1371	<b>0.0248</b>



**Fig. 12** Different sets of approximate solutions

**Fig. 13** Hypothetical decision making problem. The figure shows the 23 boxes that could be of interest for the DM for the given setting



this case from all nine components of  $P_{Q,\varepsilon}$ . Since these components are located in different regions of the parameter space, the DM is hence given a large variety for the realization of his/her project.

## 6 Conclusions and Future Work

In this chapter, we have investigated cell mapping techniques for the numerical treatment of multi-objective optimization problems. Cell mapping techniques have been designed for the global analysis of dynamical systems and replace the common point-to-point by a cell-to-cell mapping via a discretization of both space and time. We have adapted the cell mapping techniques to the given context via considering dynamical systems derived from descent methods and have argued that the resulting algorithm is in particular beneficial for the thorough investigation of small problems. That is, the new algorithm is capable of detecting the global Pareto set in one run of the algorithm which is of course important for the related decision making process. For the latter, however, also other points are of potential interest such as locally optimal solutions and approximate solutions which can serve as backup solutions for the DM in case he/she is willing to accept a certain deterioration measured in objective space. The cell mapping techniques are capable of delivering all the sets after the same run of the algorithm in the same approximation quality as the computed Pareto set. While satisfactory algorithms for the computation of the Pareto set exist, such as specialized evolutionary algorithms, this does not hold for the local and approximate solutions. The cell mapping techniques presented in this work offer hence a surplus in the design of small dimensional problems by providing a thorough analysis of the problem at hand.

Though the results presented in this work are very promising, there are some points that 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 since the number of cells grows exponentially with the number of dimensions. 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 an applicability to higher dimensional problems. Further, the constraint handling techniques may be improved so that also equality constrained problems can be treated adequately. Another interesting path of future research would be to use SCM to detect a possible bias in the descent method toward the set of interest or, if possible, to design a bias free method. This could be done by considering the volumes of the basins of attractions similar as done in [60] for general dynamical systems. Bias free methods are highly wanted for memetic strategies where the aim is to get an approximation of the entire solution set. Finally, it is expected that the change from simple cell mapping techniques as used in this chapter to generalized cell mapping will offer more information to thoroughly analyze the given model.

## References

1. Coverstone-Carroll, V., Hartmann, J.W., Mason, W.M.: Optimal multi-objective low-thrust spacecraft trajectories. *Comput. Methods Appl. Mech. Eng.* **186**(2–4), 387–402 (2000)
2. Schütze, O., Vasile, M., Coello Coello, C.A.: Computing the set of epsilon-efficient solutions in multiobjective space mission design. *J. Aerosp. Comput. Inf. Commun.* **8**, 53–70 (2011)
3. Das, I., Dennis, J.: Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM J. Optim.* **8**, 631–657 (1998)
4. Eichfelder, G.: *Adaptive Scalarization Methods in Multiobjective Optimization*. Springer, Berlin Heidelberg (2008). ISBN 978-3-540-79157-7
5. Fliege, J.: Gap-free computation of Pareto-points by quadratic scalarizations. *Math. Methods Op. Res.* **59**, 69–89 (2004)
6. Hillermeier, C.: *Nonlinear Multiobjective Optimization - A Generalized Homotopy Approach*. Birkhäuser, Boston (2001)
7. Miettinen, K.: *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, Massachusetts (1999)
8. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: multiobjective selection based on dominated hypervolume. *Eur. J. Op. Res.* **181**(3), 1653–1669 (2007)
9. Coello Coello, C., Lamont, G., Van Veldhuizen, D.: *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd edn. Springer, Berlin (2007)
10. Deb, K.: *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, Hoboken (2001)
11. Dellnitz, M., Schütze, O., Hestermeyer, T.: Covering Pareto sets by multilevel subdivision techniques. *J. Optim. Theory Appl.* **124**, 113–155 (2005)
12. Jahn, J.: Multiobjective search algorithm with subdivision technique. *Computational Optimization and Applications* **35**(2), 161–175 (2006)
13. Schütze, O., Vasile, M., Junge, O., Dellnitz, M., Izzo, D.: Designing optimal low thrust gravity assist trajectories using space pruning and a multi-objective approach. *Eng. Optim.* **41**(2), 155–181 (2009)
14. Gomez, M., Martinez-Marie, T., Sanchez, S., Meziat, D.: Optimal control for wheeled mobile vehicles based on cell mapping techniques. In: 2008 IEEE Intelligent Vehicles Symposium, pp. 1009–1014 (2008)
15. 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* **1**(3), 231–238 (2013)
16. Xiong, F.R., Qin, Z.C., Xue, Y., Schütze, O., Sun, J.Q., Ding, Q.: Multi-objective optimal design of feedback controls for dynamical systems with hybrid simple cell mapping algorithm. *Commun. Nonlinear Sci. Numer. Simul.* **19**(5), 1465–1473 (2014)
17. Zufiria, P.J., Martínez-Marín, T.: Improved optimal control methods based upon the adjoining cell mapping technique. *J. Optim. Theory Appl.* **118**(3), 657–680 (2003)
18. Hsu, C.S.: A theory of cell-to-cell mapping dynamical systems. *Journal of Applied Mechanics* **47**, 931–939 (1980)
19. Mersmann, O., Bischl, B., Trautmann, H., Preuss, M., Weihs, C., Rudolph, G.: Exploratory landscape analysis. In: GECCO, pp. 829–836 (2011)
20. Loridan, P.:  $\varepsilon$ -solutions in vector minimization problems. *Journal of Optimization, Theory and Application* **42**, 265–276 (1984)
21. 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: Emmerich M. et al. (ed.) *EVOLVE – A Bridge between Probability, Set Oriented Numerics and Evolutionary Computation IV*, pp. 171–188. Springer (2013)
22. Pareto, V.: *Cours d'Économie Politique*. Lausanne, Rouge (1896)
23. Schütze, O., Coello Coello, C.A., Talbi, E.-G.: Approximating the  $\varepsilon$ -efficient set of an MOP with stochastic search algorithms. In: Gelbukh, A., Kuri Morales, A.F. (eds.), *Mexican International Conference on Artificial Intelligence (MICAI-2007)*, pp. 128–138. Springer, Berlin Heidelberg (2007)

24. Schütze, O., Coello Coello, C.A., Tantar, E., Talbi, E.-G.: 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, pp. 713–720. ACM, New York, NY, USA (2008)
25. Hsu, C.S.: Cell-to-cell mapping: a method of global analysis for nonlinear systems. Springer-Verlag, Applied mathematical sciences (1987)
26. Gu, K., Tongue, B.H.: Interpolated cell mapping of dynamical systems. *J. Appl. Mech.* **55**(2), 461–466 (1988)
27. Guttalu, R.S., Zufiria, P.J.: The adjoining cell mapping and its recursive unraveling, part ii: application to selected problems. *Nonlinear Dyn.* **4**(4), 309–336 (1993)
28. Zufiria, P.J., Guttalu, R.S.: The adjoining cell mapping and its recursive unraveling, part i: description of adaptive and recursive algorithms. *Nonlinear Dyn.* **4**(3), 207–226 (1993)
29. Martínez-Marín, T., Zufiria, P.J.: Optimal control of non-linear systems through hybrid cell-mapping/artificial-neural-networks techniques. *Int. J. Adap. Control Signal Process.* **13**(4):307–319 (1999)
30. Bursal, F.H., Hsu, C.S.: Application of a cell-mapping method to optimal control problems. *Int. J. Control* **49**(5), 1505–1522 (1989)
31. Hsu, C.S.: A discrete method of optimal control based upon the cell state space concept. *Journal of Optimization Theory and Applications* **46**(4), 547–569 (1985)
32. Crespo, L.G., Sun, J.Q.: Stochastic Optimal Control of Nonlinear Dynamic Systems via Bellman’s Principle and Cell Mapping. *Automatica* **39**(12), 2109–2114 (2003)
33. Flashner, H., Burns, T.F.: Spacecraft momentum unloading: the cell mapping approach. *J. Guidance, Control Dyn.* **13**, 89–98 (1990)
34. Zhu, W.H., Leu, M.C.: In: Planning Optimal Robot Trajectories by Cell Mapping, pp. 1730–1735 (1990)
35. Wang, F.Y., Lever, P.J.A.: A cell mapping method for general optimum trajectory planning of multiple robotic arms. *Robot. Auton. Syst.* **12**, 15–27 (1994)
36. Yen, J.Y.: Computer disk file track accessing controller design based upon cell to cell mapping (1994)
37. Bosman, P.A.N.: On gradients and hybrid evolutionary algorithms for real-valued multiobjective optimization. *IEEE Trans. Evolut. Comput.* **16**(1), 51–69 (2012)
38. Fliege, J., Svaiter, B.F.: Steepest descent methods for multicriteria optimization. *Math. Methods Op. Res.* **51**(3), 479–494 (2000)
39. Lara, A.: Using Gradient Based Information to build Hybrid Multi-objective Evolutionary Algorithms. Ph.D thesis, CINVESTAV-IPN (2012)
40. Lara, A., Alvarado, S., Salomon, S., AviGAd, G., Coello Coello, C.A., Schütze, O.: The gradient free directed search method as local search within multi-objective evolutionary algorithms. In: EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation (EVOLVE II), pp. 153–168 (2013)
41. Coello Coello, C.A., Cruz Cortés, N.: Solving multiobjective optimization problems using an artificial immune system. *Genet. Prog. Evol. Mach.* **6**(2), 163–190 (2005)
42. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evolut. Comput.* **6**(2), 182–197 (2002)
43. Zhang, Q., Li, H.: MOEA/D: a multi-objective evolutionary algorithm based on decomposition. *IEEE Trans. Evolut. Comput.* **11**(6), 712–731 (2007)
44. Schütze, O., Esquivel, X., Lara, A., Coello, C.A.: Using the averaged Hausdorff distance as a performance measure in evolutionary multi-objective optimization. *IEEE Trans. Evolut. Comput.* **16**(4), 504–522 (2012)
45. Witting, K.: Numerical Algorithms for the Treatment of Parametric Multiobjective Optimization Problems and Applications. Ph. D. thesis, University Paderborn (2012)
46. Schütze, O.: Set Oriented Methods for Global Optimization. PhD thesis, University of Paderborn (2004). <http://ubdata.uni-paderborn.de/ediss/17/2004/schuetze/>
47. Rudolph, G., Naujoks, B., Preuss, M.: Capabilities of EMOA to detect and preserve equivalent Pareto subsets. In: EMO’03: Proceedings of the Evolutionary Multi-Criterion Optimization Conference, pp. 36–50 (2006)

48. Schäffler, S., Schultz, R., Weinzierl, K.: A stochastic method for the solution of unconstrained vector optimization problems. *Journal of Optimization, Theory and Application* **114**(1), 209–222 (2002)
49. Tanaka, M.: Ga-based decision support system for multi-criteria, optimization. In: *International Conference on Systems, Man and Cybernetics-2*, pp. 1556–1561 (1995)
50. Krmicek, V., Sebag, M.: Functional brain imaging with multi-objective multi-modal evolutionary optimization. In: Runarsson, T.P., Beyer, H.-G., Burke, E., Merelo-Guervós, J.J., Whitley, L.D., Yao, X. (eds.) *Parallel Problem Solving from Nature - PPSN IX*, volume 4193 of *Lecture Notes in Computer Science*, pp. 382–391. Springer, Berlin, Heidelberg (2006)
51. Sebag, M., Tarrisson, N., Teytaud, O., Lefevre, J., Baillet, S., Salpêtrière, L.P., Paris, F.: Multiobjective multi-modal optimization approach for mining stable spatio-temporal patterns. In: *IJCAI* (2005)
52. Tarrisson, N., Sebag, M., Teytaud, O., Lefevre, J., Baillet, S.: Multi-objective multi-modal optimization for mining spatio-temporal patterns. In: Denis, F. (ed.) *CAP*, pp. 217–230. PUG (2005)
53. Deb, K., Mohan, M., Mishra, S.: Evaluating the epsilon-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions. *Evolut. Comput.* **13**(4), 501–525 (2005)
54. Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation* **10**(3), 263–282 (2002)
55. Schütze, O., Laumanns, M., Coello Coello, C.A., Dellnitz, M., Talbi, E.: Convergence of stochastic search algorithms to finite size Pareto set approximations. *Global Optim.* **41**(4), 559–577 (2008)
56. Schütze, O., Laumanns, M., Tantar, E., Coello Coello, C.A., Talbi, E.-G.: Computing gap free Pareto front approximations with stochastic search algorithms. *Evolut. Comput.* **18**(1), 65–96 (2010)
57. Sierra, M., Coello Coello, C.A.: Improving PSO-based multi-objective optimization using crowding, mutation and  $\varepsilon$ -dominance. In: *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization*, pp. 505–519 (2005)
58. Tanaka, T.: A new approach to approximation of solutions in vector optimization problems. In: Fushini, M., Tone, K. (eds.) *Proceedings of APORS 1994*, pp. 497–504 (1995)
59. Bolintineanu, S.: (H.Bonnel). Vector variational principles;  $\varepsilon$ -efficiency and scalar stationarity. *J. Convex Anal.* **8**, 71–85 (2001)
60. Castillo, A., Zufiria, P.J.: Cell mapping techniques for tuning dynamical systems. In: Sun, J.Q., Luo, A.C.J. (eds.), *Global Analysis of Nonlinear Dynamics*, pp. 31–50. Springer (2012)