2010 7th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE 2010)
Tuxtla Gutiérrez, Chiapas, México. September 8-10, 2010.

A Predictor Corrector Method for the Computation of Boundary Points of a Multi-Objective Optimization Problem

Erick Mejía and Oliver Schütze

***Abstract*— Recently, a gradient based method has been proposed which allows to steer a given candidate solution of a multi-objective optimization problem (MOP) $F : Q \subset \mathbb{R}^n \to \mathbb{R}^k$ in any direction $\alpha \in \mathbb{R}^k$ defined in objective space. Since in the context of optimization improvements are sought, α is typically a descent direction, and the resulting curve of improving solutions steers in case the objectives are bounded below toward a boundary solution, i.e., a point x^* whose image $F(x^*)$ is at the boundary of $F(Q)$. The efficient computation of such points is of particular interest both for descent methods (i.e., to find solutions of the MOP) or for methods that move along the solution set of a MOP. Here we present a predictor corrector algorithm for the computation of such points that increases the performance of the above mentioned steering method.**

I. Introduction

In many applications, one is faced with the problem that several objectives have to be optimized concurrently leading to a multi-objective optimization problem (MOP). So far, there exists a large variety of different methods for the numerical treatment of MOPs that aim either for the approximation of the entire solution set (the so-called Pareto set), or parts of this set, or 'just' single solutions of it (see e.g. [4], [3], [2] and references therein). Recently, the Directed Search Method ([6]) has been proposed that allows to steer the search from a given solution in any direction defined in objective space. This method can either be used to improve a given candidate point (i.e., to detect single solutions) or to perform a movement along the Pareto set (i.e., to explore a part of the solution set). In both cases the subproblem is to trace a curve of candidate solutions toward boundary points of the MOP at hand. In [6], the problem is formulated as a particular initial value problem (IVP), but its numerics have not been fully exploited yet. Important in this context is to note that not the entire curve is of interest, but only the boundary point. Hence, the task is to track this curve as fast as possible to reach the end point with a low number of function or derivative evaluations of the model. Though up to date many reliable numerical methods exist for the solution of such IVPs, this reliability has to be paid in terms of function evaluations. Here we propose a specialized predictor corrector method tailored to this problem and demonstrate its superiority against the standard approach on two benchmark models.

The remainder of this paper is organized as follows: In Section II, we state the required background for the understanding of the sequel. In Section III, we present a new predictor corrector method for the computation of boundary points of a MOP. In Section IV, we present some numerical results, and finally, we conclude in Section V.

The authors are with the Departamento de Computación, CINVESTAV-IPN, Av. IPN No. 2508, Col. San Pedro Zacatenco, México, D.F. 07360, MEXICO (email: emejia@computacion.cs.cinvestav.mx, schuetze@cs.cinvestav.mx).

II. Background

In the following we consider unconstrained MOPs which are of the following form:

$$\min_{x\in\mathbb{R}^n} \{F(x)\}, \qquad \text{(MOP)}$$

where F is defined as the vector of the objective functions

$$F : \mathbb{R}^n \to \mathbb{R}^k, \qquad F(x) = (f_1(x), \ldots, f_k(x)),$$

and where each objective $f_i : \mathbb{R}^n \to \mathbb{R}$ is sufficiently smooth. The optimality of an MOP is defined by the concept of *dominance* ([5]).

Definition 1: (a) 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 $i \in \{1, \ldots, k\}$. The relation $\leq_p$ is defined analogously.

(b) A vector $y \in \mathbb{R}^n$ is *dominated* by a vector $x \in \mathbb{R}^n$ ($x \prec y$) with respect to (MOP) if

$$F(x) \leq_p F(y) \quad \text{and} \quad F(x) \neq F(y), \qquad (1)$$

else y is called non-dominated by x.

(c) A point $x \in Q$ is called *(Pareto) optimal* or a *Pareto point* if there is no $y \in Q$ which dominates x.

The set of all Pareto optimal solutions is called the *Pareto set*, and is denoted by $\mathcal{P}$. The image $F(\mathcal{P})$ of the Pareto set is called the *Pareto front*.
The derivative of F at a point x is given by

$$DF(x) = \begin{pmatrix} \nabla f_1(x)^T \\ \vdots \\ \nabla f_k(x)^T \end{pmatrix} \in \mathbb{R}^{k\times n}, \qquad (2)$$

where $\nabla f_i(x)$ denotes the gradient of objective f_i.

Recently, the *Directed Search Method* has been proposed which allows to steer the search in any direction chosen in image space ([6]). The basic idea of this approach is as follows: Assume a point $x_0 \in \mathbb{R}^n$ is given and a vector $\alpha \in \mathbb{R}^k$ representing a desired search direction in image space. To be more precise, a search direction $\nu \in \mathbb{R}^n$ in parameter space is sought such that for $y_0 := x_0 + t\nu$, where $t \in \mathbb{R}_+$ is the step size, it holds:

$$\lim_{t\searrow 0} \frac{f_i(y_0) - f_i(x_0)}{t} = \langle \nabla f_i(x_0), \nu\rangle = \alpha_i, \quad i = 1, \ldots, k \qquad (3)$$

Using the Jacobian of F, Equation (3) can be stated in matrix vector notation as

$$DF(x_0)\nu = \alpha. \tag{4}$$

Hence, such a search direction ν can be computed by solving a system of linear equations. Since typically the number of parameters is (much) higher than the number of objectives in a given MOP, i.e., $n >> k$, system (4) is (probably highly) underdetermined which implies that its solution is not unique. To prevent this, the solution with the lowest norm can be chosen leading to

$$\nu = DF(x_0)^+\alpha, \tag{5}$$

where $A^+ \in \mathbb{R}^{n\times k}$ denotes the pseudo inverse of a matrix $A \in \mathbb{R}^{k\times n}$, $k \le n$. In case the rank of A is maximal (i.e., k), the pseudo inverse is given by $A^+ = A^T(AA^T)^{-1}$. This result can be used to define a curve $x(t) \subset \mathbb{R}^n$, $t \in [0,T]$, of points such that its image curve $F(x(t)) \subset \mathbb{R}^k$ is a line segment from $F(x_0)$ pointing in direction α (compare to Figure 1).

Using

$$\nu_\alpha(x) := -DF(x)^+\alpha, \tag{6}$$

the curve $x(t)$ can be obtained by numerically solving the following initial value problem:

$$\begin{aligned} x(0) &= x_0 \in \mathbb{R}^n \\ \dot{x}(t) &= \nu_\alpha(x(t)), \quad t \in [0,T]. \end{aligned} \tag{IVP$_\alpha$}$$

In certain applications one is interested in computing the set

$$H(x) = 0 \tag{7}$$

If, for instance, $H : \mathbb{R}^{N+1} \to \mathbb{R}^N$ is a sufficiently smooth mapping and $x_0 \in \mathbb{R}^{n+1}$ a point that satisfies (7), then one can expect that $H^{-1}(0)$ defines at least around x_0 a curve. To track such a curve numerically, predictor-corrector (PC) methods can be chosen. Starting with a given solution of (7), the next one is obtained in two steps: First, a rough approximation p is computed e.g. via a linearization of the curve. In the second step, this predictor solution p is corrected back to $H^{-1}(0)$.

III. A PC Method for the Numerical Solution of (IVP$_\alpha$)

The question is how to solve (IVP$_\alpha$) efficiently. Note that we are only interested in the end point of that curve (which is hopefully a Pareto optimal solution), hence, in order to save function evaluations, large step sizes along the curve are desired.

In the following we propose one possible way to compute the solution curve numerically by a PC method. The key observation for this is that the image points of the solution curve are (by construction) located on a half line from $F(x_0)$ pointing in direction α (see Figure 2). That is, for every point x on the solution curve it holds

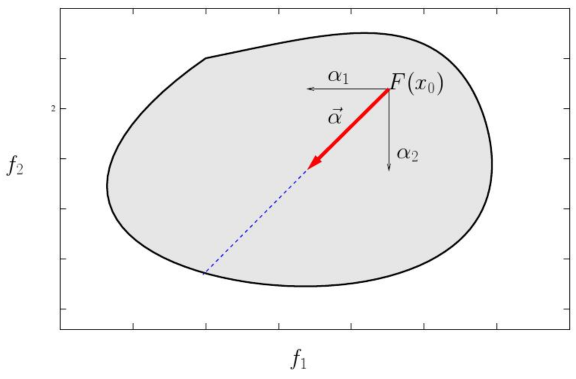

Fig. 1. The basic idea in the *Directed Search Method* is to generate a curve of dominating points given a desired direction $\alpha \in \mathbb{R}^k$ and a point $x_0 \in \mathbb{R}^n$.

$$F(x) = F(x_0) + \lambda_y\alpha, \tag{8}$$

where λ_y is nonnegative. Hence, the curve is contained in the zero set of

$$\begin{aligned} H &: \mathbb{R}^{n+1} \to \mathbb{R}^k \\ H(x,\lambda_y) &= F(x) - F(x_0) - \lambda_y\alpha \end{aligned} \tag{9}$$

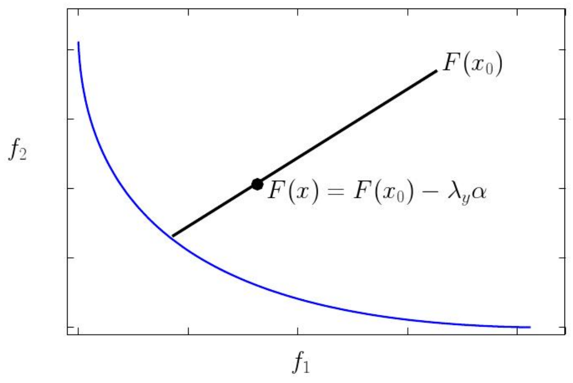

Fig. 2. Every point in the solution curve (image space) can be parametrized by λ_y.

To comply with the needs of PC methods we introduce an additional parameter λ_x into the solution curve which can be expressed as the solution of the following auxiliary IVP (note that λ_x is defined in parameter space):

$$\begin{aligned} x(0) &= (x_0, \lambda_{x,0} = 0) \in \mathbb{R}^{n+1} \\ \dot{x}(t) &= \begin{pmatrix} \nu_\alpha(x(t)) \\ 1 \end{pmatrix}, \quad t > 0 \end{aligned} \tag{IVP$_{\alpha,\lambda}$}$$

So far, we are not able to apply PC methods since the parameters λ_x and λ_y parametrize different curves. The following little consideration, however, argues that it is reasonable to match the two parameters: Let x_0 be given and $\lambda_{x,0} = 0$, and let $(x_1, \lambda_{x,1})$ be an Euler step of (IVP$_{\alpha,\lambda}$) with a small step size $\Delta\lambda_x$, i.e.,

$$x_1 = x_0 + \Delta\lambda_x \nu_\alpha(x_0), \quad \lambda_{x,1} = \Delta\lambda_x. \tag{10}$$

By construction of $\nu_\alpha(x_0)$ and since $\Delta\lambda_x$ is small we have

$$\alpha_i \approx \frac{f_i(x_1) - f_i(x_0)}{\Delta\lambda_x}, \quad i = 1, \dots, k. \tag{11}$$

This implies that $F(x_1) - F(x_0) \approx \Delta\lambda_x\alpha$ and hence $H(x_1, \lambda_{x,1}) \approx 0$ which suggests to make the substitution

$$\lambda_x = \lambda_y \tag{12}$$

Using this, (IVP$_{\alpha,\lambda}$) and (9) can now be used to perform classical PC methods in order to trace a solution curve: starting with the point $(x_0, \lambda_{x,0})$ one can integrate (IVP$_{\alpha,\lambda}$) numerically for a small time step, e.g. via the Euler method as described above leading to a predictor solution $(\tilde{x}_1, \tilde{\lambda}_{x,1})$. In a next step this solution can be corrected to the desired curve (in objective space). That is, starting with $(\tilde{x}_1, \tilde{\lambda}_{x,1})$ and using a root finding method applied on (9) one can try to find a solution $(x_1, \lambda_{x,1})$ with $H(x_1, \lambda_{x,1}) \approx 0$, and so on. Since it is desired to obtain the closest solution of H to $(\tilde{x}_1, \tilde{\lambda}_{x,1})$, it is recommended to take the (Gauss-) Newton method for the corrector step ([1]). By the above consideration, $\lambda := \lambda_y$ can be chosen as the pseudo arc length to trace such a curve.

It remains to compute the step size control for the PC method. Most influential for the distance between consecutive solutions on a curve is the choice of the step size for the predictor. Here we take advantage of our particular setting and suggest a step size that is based on the curvature of the curve at the current point x (i.e., in parameter space) as well as the deviation of $F(x) - F(p)$ to the desired direction α (in objective space).
In the first step we calculate a maximal step size where the predictor is expected to be in the ϵ neigborhood of the zero set of H based on the estimation of the curvature $\kappa(\lambda) = \|\gamma''(\lambda)\|$ of the numerical approximation of the 'curve' $\gamma(\lambda)$ of (IVP$_{\alpha,\lambda}$) (note that we do not have a representation of this curve, just sample points along it). Assume we are given the current iterate (x_k, λ_k), and have already computed the two predesessors x_{k-1} and x_{k-2} (else we use a predefined step size control). For the parametrization we can choose again λ_y as described above. Hence, we are given the three points along the discretized curve

$$\gamma(\lambda_i) = x_i, \quad i = k, k-1, k-2. \tag{13}$$

We can now estimate the curvature by applying twice the backward difference quotient. Using this, the approximation of the i-th component of the second derivative $\gamma(\lambda_k)$ at the point λ_k reads as follows:

$$\gamma''(\lambda_k)_i \approx \frac{\frac{\gamma(\lambda_k)_i - \gamma(\lambda_{k-1})_i}{\lambda_k - \lambda_{k-1}} - \frac{\gamma(\lambda_{k-1})_i - \gamma(\lambda_{k-2})_i}{\lambda_{k-1} - \lambda_{k-2}}}{\lambda_k - \lambda_{k-1}} \tag{14}$$

.

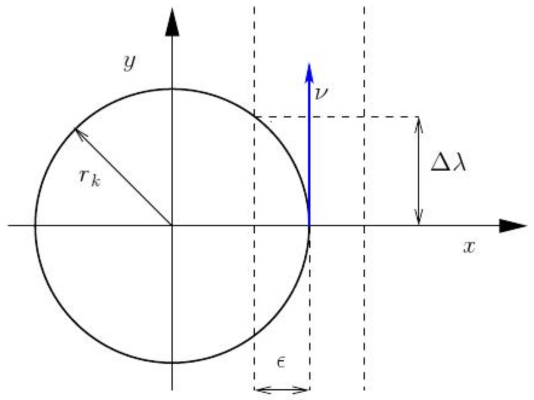

Fig. 3. When assuming that the curvature κ is fixed the maximal step size $\Delta\lambda$ can be computed by trigonometric equations. Here, ν represents the idealized search direction, and the circle the solution curve with constant curvature.

Using $\kappa(\lambda_k)$, the initial size for the predictor is chosen as follows: We allow a tolerance value of ϵ in parameter space, i.e., $\|H(x, \lambda)\| \le \epsilon$, where we fix $\lambda = \lambda_k$ and where $\epsilon \in \mathbb{R}_+$.

Using $r_k = 1/\kappa(\lambda_k)$, we estimate the step length where this tolerance is violated, in particular, we use (compare to Figure 3):

$$\begin{aligned} \Delta\lambda_k &:= \sqrt{{r_k}^2 - (r_k - \epsilon)^2} \\ &= \sqrt{2r_k\epsilon - \epsilon^2}, \end{aligned} \tag{15}$$

in case this value is within a prediscribed range $[\Delta_{min}, \Delta_{max}]$, else we choose the respective boundary value.

In the second step we refine the step length by considering the objective value of the predictor point. Given a tolerance value β (see picture 4), we consider $\Delta\lambda_y = \Delta\lambda_k$ as a valid step size if the angle between $F(x_i) - F(x_{i-1})$ and $F(x_{i-1}) - \alpha$ is less or equal than β. If the current value of $\Delta\lambda_y$ violates the cone we propose to make a backtracking correction, i.e.,

$$\lambda_y = \rho\lambda_y\,, \quad \rho \in (0, 1) \tag{16}$$

In Algorithm 1 we summarize the computation of an 'acceptable' step predictor step size.

Algorithm 1: **Computation of the Predictor Step Size**

```
Input: $\beta$, $x_i$, $\rho \in (0,1)$
Output: $\Delta\lambda_y$
1 compute $\Delta\lambda_k$ as in (15);
2 $\lambda_y \leftarrow \Delta\lambda_k$;
3 $\nu \leftarrow DF(x)^+\alpha$;
4 repeat
5     $y \leftarrow x + \lambda_y\nu$;
6     $F_x \leftarrow F(x)$, $F_y \leftarrow F(y)$;
7     if $\measuredangle(F_x - \alpha,\ F_x - F_y) > \beta$ then $\Delta\lambda_y \leftarrow \rho\Delta\lambda_y$
8 until $\measuredangle(F_x - \alpha,\ F_x - F_y) \le \beta$ ;
9 return $\Delta\lambda_y$
```

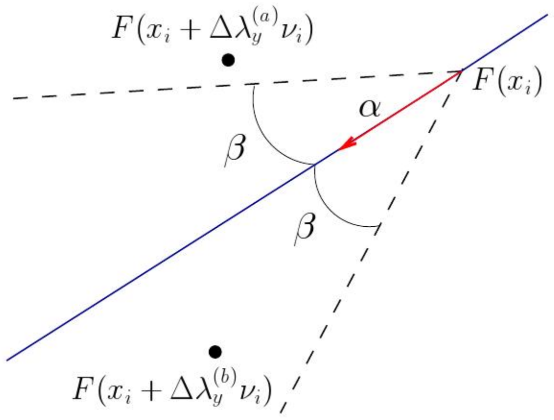
$F(x_i + \Delta\lambda_y^{(a)} \nu_i)$
$F(x_i)$
α
β
β
$F(x_i + \Delta\lambda_y^{(b)} \nu_i)$

Fig. 4. The tolerance cone is defined by a given tolerance β. In this case, $\Delta\lambda_y^{(b)}$ would be considered as a valid step size and we are able to compute the corrector. On the other hand, $\Delta\lambda_y^{(a)}$ would be rejected and we have to make a backtracking correction.

IV. Numerical Results

In the following we test our new approach on two bi-objective optimization problems. We compare our results with the algorithms ode23 and ode45 of MATLAB[1] for the solution of IVPs. Since in all methods the number of Jacobian calls of the MOP is deciding for the computational time, we compare only these values.

A. Example 1

First we consider the following convex model

$$\min F : \mathbb{R}^n \to \mathbb{R}^2$$
$$f_i(x) = \|x - a_i\|_2^2 = \sum_{j=1}^{n} (x_j - a_{i,j})^2, \tag{17}$$

where $a_{i,j}$ denotes the j-th entry of a given vector a_i, and

$$\begin{aligned} a_1 &= \{1, \dots, 1\} \in \mathbb{R}^n \\ a_2 &= \{-1, \dots, -1\} \in \mathbb{R}^n \end{aligned} \tag{18}$$

Figure 5 shows one numerical result from the PC method with a given value x_0 and for the dimension $n = 100$ of MOP (17). For the predictor step, $\beta = 5$. Table I shows a comparison of the number of required Jacobian calls of the three different algorithms averaged over 100 runs with different starting points within the domain $Q = [-5, 5]^{100}$. Each run was terminated if the 2-condition of the Jacobian in (4) exceeded the value 10^7 ([6]). The PC method is most efficient in this setting.

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

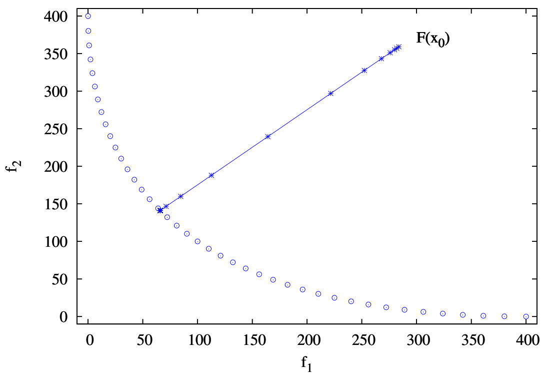

Fig. 5. Numerical result of the PC method for MOP (17) for $n = 100$ and direction $\alpha = (-1, -1,)^T$. Here, 11 steps were required to reach a boundary point.

TABLE I
NUMBER OF REQUIRED DERIVATIVE CALLS FOR MOP (17) FOR $n = 100$. THE RESULTS ARE AVERAGED OVER 100 DIFFERENT STARTING POINTS WITHIN $Q = [-5, 5]^{100}$.

Algorithm	Number of DF calls
PC	**16**
ode23	72
ode45	418

B. Example 2

Our second example is a non-convex problem ([7]):

$$\begin{aligned} & f_1, f_2 : \mathbb{R}^2 \to \mathbb{R} \\ & f_1(x, y) = \frac{1}{2}\left(\sqrt{1 + (x + y)^2} + \sqrt{1 + (x - y)^2} + x - y\right) \\ & \qquad + \lambda \cdot e^{(-x-y)^2} \\ & f_2(x, y) = \frac{1}{2}\left(\sqrt{1 + (x + y)^2} + \sqrt{1 + (x - y)^2} - x + y\right) \\ & \qquad + \lambda \cdot e^{(-x-y)^2} \end{aligned} \tag{19}$$

where we have chosen $\lambda = 0.85$. For the predictor step, $\beta = 5$. One particular result is shown in Figure 6, and the averaged results are shown in Table II. Here, we have stopped the process if the 2-condition number of 10^5 has been reached (note the smaller dimension of the domain in this problem). Due to the non-convexity of the MOP, the two IVP solvers need a large amount of Jacobian calls before detecting the boundary point, and the difference to the performance of the PC method becomes even more significant.

Finally, we apply the continuation method proposed in [6] to compute the Pareto set of MOP (19). In this method, one problem of the form (IVP$_\alpha$) has to be solved for the computation of each new candidate solution along the Pareto front (which serves as the corrector step in this case). Here, we compare this method to the variation where we use the novel PC method for the computation of the respective boundary point. Using $\epsilon = 1 \times 10^{-3}$ we obtain for both algorithms

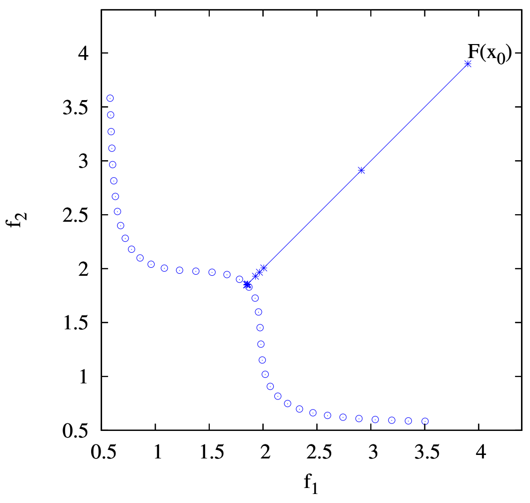

Fig. 6. Numerical result of the PC method for MOP (19) and direction vector $\alpha = (-1,-1)^T$. Here, 7 steps were required to reach a boundary point.

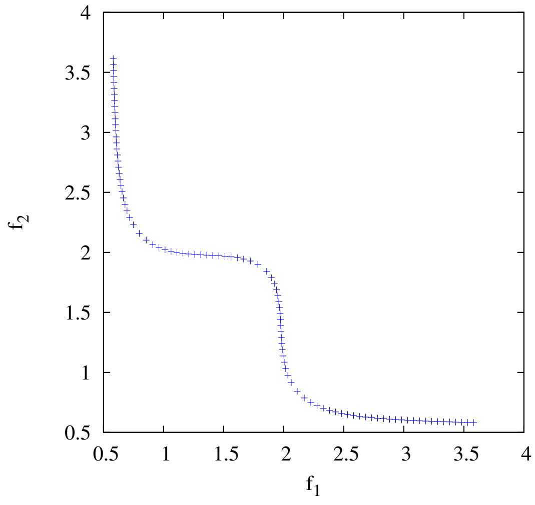

Fig. 7. Numerical result of a continuation method for MOP (19), using the Directed Search Method to solve the subproblems.

TABLE II
NUMBER OF REQUIRED DERIVATIVE CALLS FOR MOP (19). THE RESULTS ARE AVERAGED OVER 100 DIFFERENT STARTING POINTS WITHIN $Q = [-5,5]^2$.

Algorithm	Number of DF calls
PC	**12**
ode23	276
ode45	503

the result shown in Figure 7. However, the computational time differs since the number of corrector steps used differs significantly for both methods. In the classical method, a total of 1056 Jacobian calls were required (which makes in average 11 Jacobian calls to obtain the 96 elements shown in Figure 7). In contrast, when using the new PC variant for the corrector, merely 193 Jacobian were required (which leads to an average result of 2 Jacobian calls per corrector step).

V. CONCLUSIONS

Here we have proposed a specialized predictor corrector method for the detection of boundary points of a multi-objective optimization problem based on the Directed Search Method [6]. The method is an improvement of the approach suggested in [6] since it does not focus on the approximation of the entire curve, but aims to find large step sizes in order to reach the end point of a curve rapidly. We have demonstrated the superiority of the novel approach on two benchmark models.

Predictor Corrector Method proposed by the authors is a very interesting idea to obtain movement vectors in variable space from a direction vector in objective space. Although the method needs the compute of the gradient/jacobian matrix, it can be useful (very fast) for small/simple problems.

REFERENCES

[1] E. L. Allgower and K. Georg. *Numerical Continuation Methods*. Springer, 1990.
[2] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, second edition, 2007.
[3] M. Ehrgott. *Multicriteria Optimization*. Springer, 2005.
[4] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, Massachusetts, 1999.
[5] Vilfredo Pareto. *Manual of Political Economy*. The MacMillan Press, 1971 (original edition in French in 1927).
[6] O. Schütze, A. Lara, and C. A. Coello Coello. The directed search method for unconstrained multi-objective optimization problems. Technical report TR-OS-2010-01, see http://delta.cs.cinvestav.mx/schuetze/technical_reports/, CINVESTAV-IPN, 2010.
[7] K. Witting and M. Hessel von Molo. Private communication, 2006.