

A generalized computing paradigm based on artificial dynamic models for mathematical programming

F. Torelli · A. Vaccaro

Published online: 26 October 2013
© Springer-Verlag Berlin Heidelberg 2013

Abstract In this paper a novel computing paradigm aimed at solving non linear systems of equations and finding feasible solutions and local optima to scalar and multi objective optimizations problems is conceptualized. The underlying principle is to formulate a generic programming problem by a proper set of ordinary differential equations, whose equilibrium points correspond to the problem solutions. Starting from the Lyapunov theory, we will demonstrate that this artificial dynamic system could be designed to be stable with an exponential asymptotic convergence to equilibrium points. This important feature allows the analyst to overcome some of the inherent limitations of the traditional iterative solution algorithms that can fail to converge due to the highly nonlinearities of the first-order conditions. Besides we will demonstrate as the proposed paradigm could be applied to solve non linear equations systems, scalar and multi-objective optimization problems. Extensive numerical studies aimed at assessing the effectiveness of the proposed computing paradigm are presented and discussed.

Keywords Dynamic system theory · Artificial models · Non linear systems of equations · Multiobjective · Programming problems

Communicated by E. Munoz.

F. Torelli
Department of Electrical Engineering, Politecnico di Bari,
70125 Bari, Italy

A. Vaccaro (✉)
Department of Engineering, Università degli studi del Sannio,
82100 Benevento, Italy
e-mail: vaccaro@unisannio.it

1 Introduction

Mathematical programming deals with the strategic allocation of limited resources among competing activities, in the presence of a set of constraints imposed by the problem domain being analyzed. The problem constraints could derive by financial, technological, social or many other considerations.

Mathematical programming problems can be formalized by systems of nonlinear equations, scalar and multi-objective optimization problems that are solved by different and specific solution strategies depending of the nature of the problem and the application domain.

In particular, many Newton-type algorithms have been proposed in the literature for solving systems of nonlinear equations (Malick and Roupin 2013). The insight is to approximate the non-linear equations by linearized Jacobian-matrix equations and solve them by means of numerical iteration algorithms. In addressing this issue the adoption of sparse triangular factorization techniques aimed at improving the algorithm performance is frequently adopted (Tinney and Walker 1967). More sophisticated solution approaches include the trust-region method (Conn et al. 2000), the Broyden (1965) method, the secant method (Denis and Wolkowicz 1993) and the Halley method (Ortega and Rheinboldt 1970).

Despite their stability and their powerful convergence properties, these solution approaches could be unexceptionally restricted by a non-singular Jacobian matrix (namely the matrix of all first-order partial derivatives of the system of equations). In particular they could become instable or even completely fail when the attraction region of the initial solution guess is far away from the problem solution. These issues are not infrequent in real world applications since the problem is nondeterministic polynomial-time hard, and it could

be characterized by very high computational complexity due to many numerical issues (Grosan and Abraham 2008).

As far as the scalar optimization problems are concerned, they consist of maximizing/minimizing a scalar function (a.k.a. the objective function) and satisfying a number of constraints or limitations. A feasible solution that minimizes/maximizes the objective function is called an optimal solution. Generally, unless both the objective function and the feasible region are convex, there may be several local minima/maxima.¹ Scalar optimization problems can be solved by different solution techniques as far as nonlinear programming (Zeshui et al. 2011), quadratic programming (Delbos and Gilbert 2005), and linear programming (Martinez 1994) are concerned. Some algorithms formalize the problem's Karush–Kuhn–Tucker optimality conditions which are a set of nonlinear equations that can be solved by using iterative Newton-based algorithms. These methods can handle both equality and inequality constraints. The latter can be added as quadratic penalty terms to the objective function and multiplied by proper penalty multipliers (Li and Gen 1996). Another useful paradigm for inequality constraints handling is based on the Interior Point method (a.k.a. barrier method) (Astfalk et al. 1992). The insight is to convert the inequality constraints into equalities by the introduction of nonnegative slack variables. A self-concordant barrier function of these slack variables² is then added to the objective function and multiplied by a barrier parameter (which is gradually reduced to zero during the solution process).

These programming algorithms represent a useful tool for solving scalar optimization problems but, as evidenced by the many discussions reported in the literature (Michalewicz 1995), they could reveal some shortcomings principally arising from the limited capability to solve real-world large-scale problems, the weakness in handling qualitative constraints (namely constraints defined by information granules), the poor convergence in computing global optimums, the difficulties in addressing ill conditioned problems.

All these scalar optimization algorithms are in principle not suitable for addressing constrained optimization problems involving more than one objective function to be optimized simultaneously (a.k.a. multiobjective programming problems). The main difficulties derive by the multiplicity of

the problem objectives conflicting across a high-dimension search space. To address this problem proper optimality criteria should be defined. To this aim many solution techniques refer to the concept of Pareto optimality which aims at identifying proper trade-offs between the problem objectives (Narula et al. 1994). In details, a solution is called Pareto optimal (a.k.a. non-dominated solution) if none of the objective functions can be improved in value without degrading some of the other objective values. Without an additional external preference criteria, all Pareto optimal solutions (namely the Pareto front) could be considered equally acceptable by the analyst. Consequently solving a multi-objective optimization problem asks for computing all or a representative set of Pareto optimal solutions. Anyway, these solutions cannot be computed efficiently in many cases since they are often of exponential size and NP-hard to compute. As a consequence, approximation methods are frequently used.

The simpler algorithm aimed at approximating the Pareto front is the so called weighting strategy (a.k.a. scalarization method) (Park 2004). This approach aims at combining the multiple problem objectives into one single-objective scalar function. To this aim a positively weighted convex sum of the objectives is typically adopted. By varying these weights it is possible to obtain Pareto optimal solutions but only for problems characterized by convex Pareto fronts.

To overcome these limitations, more sophisticated algorithms could be adopted. They include ε -constraints Method (Chinchuluun and Pardalos 2007), Goal Programming (Ng 1987), Multi-level Programming (Okabe et al. 2003).

These methods allow an effective exploration of the solution space but they require an a priori knowledge of the problem being analyzed and they could be sensitive to the shape of the Pareto front (Marler and Arora 2004).

A promising research direction aimed at overcoming some of these limitations is to rely on Metaheuristic algorithms [i.e. Genetic Algorithms (Fonseca and Fleming 1998), Evolutionary Programming (Rodriguez-Vazquez et al. 2004), Particle Swarm Optimization (Agrawal et al. 2008)]. A review of the wide literature on the role of bio-inspired and population based methods for multi-objective optimization is presented in Coello et al. (2007). As outlined in these papers, metaheuristic algorithms allow the analyst to effectively handle qualitative constraints, to improve the solution space exploration and to reduce the probability to converge on local minima. Despite these benefits, mathematical analysis of these algorithms lacks behind. In particular, convergence analysis still remains almost unsolved, while computation burden is equally challenging (Yang 2011).

From the analysis of these papers, a number of important scientific and technical points comes out, stimulating the conceptualization of alternative computing paradigms for the effective solution of mathematical programming problems.

¹ A local minimum/maxima x^* is defined as a point satisfying the following condition:

$$\exists \delta > 0' \|x - x^*\| < \delta \Rightarrow \begin{cases} f(x^*) \leq f(x) & \text{Local Minima} \\ f(x^*) \geq f(x) & \text{Local Maxima} \end{cases}$$

² An example of barrier function is the so called logarithmic barrier function defined as:

$$B(x, \mu) = f(x) - \mu \sum_{i=1}^m \ln(c_i(x))$$

where $f(x)$ is the objective function, μ is the slack variable and $c_i(x)$ is the i -th the inequality constraint function.

In details, it is worth observing as the discussed solution techniques: (1) can be effectively addressed to solve a specific programming problem; (2) could fail to converge in the presence of singularities of the Jacobian matrix; (3) could become instable when the attraction region of the initial solution guess is far away from the problem solution. To overcome these limitations in this paper a novel computing paradigm is conceptualized.

The proposed solution is based on a challenging idea, originated from papers Xie et al. (2013), Torelli et al. (2013) and based on the dynamic systems theory. The idea is to formulate a generic mathematical programming problem by a set of ordinary differential equations, whose equilibrium points represent the problem solutions. Staring form the Lyapunov theory, we will demonstrate that this artificial dynamical model is asymptotically stable and it is quite insensitive to many factors which can cause numerical instabilities to the traditional solution algorithms. This important feature allows the analyst to overcome the inherent limitations of iterative algorithms that can fail to converge due to the highly nonlinearities of the first-order conditions.

This paper intends to bring the following contributions to the literature:

1. First of all we conceptualize a generalized computational paradigm aims at solving non linear systems of equations, scalar and multi objective optimizations problems;
2. We theoretically characterize its convergence proprieties;
3. We demonstrate as the proposed paradigm converges to a root of a system of equations (when applied to non linear systems solution), to a feasible solution which locally minimizes the objective function (when applied to scalar optimization problems) or to a feasible Pareto solution (when applied to multi objective optimization problems);

Detailed numerical experimental results obtained by applying the proposed paradigm on several benchmarks will be presented and discussed in order to prove its effectiveness in managing complex and real world problems.

2 Theoretical foundation

2.1 Non linear equations systems

The mathematical kernel of the proposed computing paradigm is based on the solution of the following nonlinear system of equations:

$$\begin{cases} g_1(\mathbf{x}) = 0 \\ g_2(\mathbf{x}) = 0 \\ \dots \\ g_N(\mathbf{x}) = 0 \end{cases} \quad (1)$$

where $g_i : \mathbf{R}^n \rightarrow \mathbf{R}, i = 1, 2, \dots, N$ are continuously differentiable functions on \mathbf{gR}^n .

The most common approach aimed at solving the problem (1) minimizes the sum of the squared residuals. This allow us to formalize the problem (1) as follow:

$$\begin{cases} \min_{\mathbf{x}} w(\mathbf{x}) \\ w(\mathbf{x}) = \frac{1}{2} \sum_{i=1}^N g_i(\mathbf{x})^2 = \frac{1}{2} \mathbf{g}(\mathbf{x})^T \mathbf{g}(\mathbf{x}) \end{cases} \quad (2)$$

where $\mathbf{g}(\mathbf{x})^T = [g_1(\mathbf{x}), \dots, g_N(\mathbf{x})]$ and T is the transposition operator.

To solve the problem (2) the traditional solution approaches try to solve the first-order derivative condition:

$$\frac{dw(\mathbf{x})}{d\mathbf{x}} = \left(\frac{d\mathbf{g}(\mathbf{x})}{d\mathbf{x}} \right)^T \mathbf{g}(\mathbf{x}) = 0 \quad (3)$$

The solution of this problem can be impracticable due to the nonlinear nature of the resulting set of equations, so numerical methods are traditionally employed in order to obtain a solution that is within an acceptable tolerance.

In this paper we propose an alternative computing paradigm aimed at solving the problem (2). The underlying principle is to design a stable artificial dynamic system whose equilibrium points represent the solutions of the problem (1). In particular, we assume that the components of the vector \mathbf{x} evolve according to proper time functions $\mathbf{x}(t)$.³ Under this assumption, the scalar positive-semidefinite objective function $w(\mathbf{x})$ can be considered as a Lyapunov function. Consequently if we let the time derivative of $w(\mathbf{x})$ negative-definite or negative-semidefinite along the trajectory $\mathbf{x}(t)$, then the Lyapunov theorem would assure the existence of an asymptotically stable equilibrium point which minimizes $w(\mathbf{x})$ and solves the problem (1).

In order to prove this statement, let's compute the time derivative of $w(\mathbf{x})$ by applying the chain rule:

$$\frac{dw(\mathbf{x}(t))}{dt} = \dot{w}(\mathbf{x}(t)) = \mathbf{g}(\mathbf{x}(t))^T \frac{d\mathbf{g}(\mathbf{x}(t))}{dt} = \mathbf{g}(\mathbf{x}(t))^T \dot{\mathbf{g}}(\mathbf{x}(t)) \quad (4)$$

And since:

$$\dot{\mathbf{g}}(\mathbf{x}(t)) = \frac{d\mathbf{g}(\mathbf{x}(t))}{d\mathbf{x}} \frac{d\mathbf{x}(t)}{dt} = \frac{d\mathbf{g}(\mathbf{x}(t))}{d\mathbf{x}} \dot{\mathbf{x}}(t) \quad (5)$$

we obtain:

$$\dot{w}(\mathbf{x}(t)) = \mathbf{g}(\mathbf{x}(t))^T \frac{d\mathbf{g}(\mathbf{x}(t))}{d\mathbf{x}} \dot{\mathbf{x}}(t) \quad (6)$$

³ It is worth noting as the variable "t" is an artificial parameter and we are only interested in the final equilibrium points reached by the dynamic system and the transient nature of the trajectories.

Consequently if we let $\dot{\mathbf{x}}(t)$ change according to the gradient of $w(\mathbf{x})$:

$$\dot{\mathbf{x}}(t) = -k \left(\frac{dw(\mathbf{x})}{d\mathbf{x}} \right)^T = -k \left(\frac{d\mathbf{g}(\mathbf{x})}{d\mathbf{x}} \right)^T \mathbf{g}(\mathbf{x}) \tag{7}$$

where the gradient is a row vector and k is a positive constant, we obtain:

$$\dot{w}(\mathbf{x}(t)) = -k \left[\mathbf{g}(\mathbf{x}(t))^T \frac{d\mathbf{g}(\mathbf{x}(t))}{d\mathbf{x}} \left(\frac{d\mathbf{g}(\mathbf{x}(t))}{d\mathbf{x}} \right)^T \mathbf{g}(\mathbf{x}(t)) \right] \tag{8}$$

This is a quadratic form which is certainly negative-semidefinite.

This important result allows us to conclude that if the vector $\mathbf{x}(t)$ evolves according to (7), then the following Lyapunov conditions are satisfied:

1. $w(\mathbf{x})$, is positive-semidefinite and equals to zero at the equilibrium points;
2. the time derivative of $w(\mathbf{x})$ is negative-semidefinite;

Consequently $\mathbf{x}(t)$ converges to an equilibrium point \mathbf{x}^* which is asymptotically stable⁴

Moreover, by substituting the (7) into (5) we obtain:

$$\dot{\mathbf{g}}(\mathbf{x}(t)) = -k \frac{d\mathbf{g}(\mathbf{x}(t))}{d\mathbf{x}} \left(\frac{d\mathbf{g}(\mathbf{x})}{d\mathbf{x}} \right)^T \mathbf{g}(\mathbf{x}) \tag{9}$$

By observing that the matrix $\left(\frac{d\mathbf{g}(\mathbf{x})}{d\mathbf{x}} \left(\frac{d\mathbf{g}(\mathbf{x})}{d\mathbf{x}} \right)^T \right)$ is symmetric and positive-definite and its eigenvalues are real and positive, we can conclude that $\mathbf{g}(\mathbf{x}(t))^T = [g_1(\mathbf{x}(t)), \dots, g_N(\mathbf{x}(t))]$ exponentially converges to the equilibrium point $\mathbf{g}(\mathbf{x}^*) = 0$.

This important result allows us to solve the problem (1) by the time simulation of the artificial dynamic system (7) without requiring any matrix inversion and/or factorization and overcoming the main difficulties arising by the ill-conditioning of the Jacobian matrix.

2.2 Scalar optimization problems

In this section we will show as the previously described dynamic paradigm can be effectively deployed to solve the following constrained nonlinear programming problem:

$$\begin{cases} \min_{\mathbf{x}} f(\mathbf{x}) \\ s.t. \quad \mathbf{g}(\mathbf{x}) = 0 \\ \mathbf{h}_{\min} \leq \mathbf{h}(\mathbf{x}) \leq \mathbf{h}_{\max} \end{cases} \tag{10}$$

⁴ Since the system of differential equations (7) is nonlinear, the corresponding equilibrium point depends of the initial condition. This implies that the dynamic model could have different equilibrium points for different initial conditions. Further studies aiming at characterizing the dimension of the regions of attraction is currently under investigation by the authors.

where \mathbf{x} is the vector of the control/decision variables, $\mathbf{g}(\mathbf{x})$ is the n -dimensional equality constraint vector and $\mathbf{h}(\mathbf{x})$ is the m -dimensional vector describing the inequality constraints.

To solve this problem by the proposed computing paradigm it is worth observing as:

1. the minimization of the objective function $f(\mathbf{x})$ is equivalent to find the zeros of the following function:

$$F_1(\mathbf{x}, q) = f(\mathbf{x}) - q \tag{11}$$

where q is an additional slack variable that allows us to satisfy the conditions imposed by the Lyapunov theorem on points different from the zeros of the objective function.

2. the satisfaction of the equality constraints requires the roots finding of the following functions:

$$F_2(\mathbf{x}) = \mathbf{g}(\mathbf{x}) \tag{12}$$

3. the inequality constraints can be converted into equality constraints by the introduction of nonnegative slack variables according to the Interior Point theory (Astfalk et al. 1992; Boggs et al. 1996). Consequently their satisfaction asks for finding the zeros of the following functions:

$$\begin{cases} F_3(\mathbf{x}, \mathbf{s}) = \mathbf{h}(\mathbf{x}) + \mathbf{s} - \mathbf{h}_{\max} \\ F_4(\mathbf{x}, \mathbf{t}) = \mathbf{h}(\mathbf{x}) - \mathbf{t} - \mathbf{h}_{\min} \end{cases} \tag{13}$$

where \mathbf{s} and \mathbf{t} are two additional unknown vectors representing nonnegative slack variables.

Thanks to these statements the scalar optimisation problem formalised in (10) can be addressed by solving the following nonlinear system of equations:

$$\begin{cases} F_1(\mathbf{x}, q) = 0 \\ F_2(\mathbf{x}) = 0 \\ F_3(\mathbf{x}, \mathbf{s}) = 0 \\ F_4(\mathbf{x}, \mathbf{t}) = 0 \end{cases} \tag{14}$$

Consequently, our goal is to compute the value of the vector $\mathbf{z} = [\mathbf{x}, q, \mathbf{s}, \mathbf{t}]$ that simultaneously makes equal to zero each component of the vector $\mathbf{F}(\mathbf{z}) = [F_1(\mathbf{x}, q), F_2(\mathbf{x}), F_3(\mathbf{x}, \mathbf{s}), F_4(\mathbf{x}, \mathbf{t})]$.

The solution of this problem by the proposed computing paradigm requires the time simulation of the following dynamic system:

$$\dot{\mathbf{z}} = -k \left(\frac{d\mathbf{F}(\mathbf{z})}{d\mathbf{z}} \right)^T \mathbf{F}(\mathbf{z}) \tag{15}$$

which can be rewritten as follows:

$$\begin{cases} \dot{\mathbf{x}} = -k_x \left[\left(\frac{df(\mathbf{x})}{d\mathbf{x}} \right)^T (f(\mathbf{x}) - q) + \left(\frac{d\mathbf{g}(\mathbf{x})}{d\mathbf{x}} \right)^T \mathbf{g}(\mathbf{x}) \right. \\ \left. + \left(\frac{d\mathbf{h}(\mathbf{x})}{d\mathbf{x}} \right)^T (\mathbf{h}(\mathbf{x}) + \mathbf{s} - \mathbf{h}_{\max}) + \left(\frac{d\mathbf{h}(\mathbf{x})}{d\mathbf{x}} \right)^T (\mathbf{h}(\mathbf{x}) - \mathbf{t} - \mathbf{h}_{\min}) \right] \\ \dot{\mathbf{s}} = -k_s \mathbf{I}(\mathbf{h}(\mathbf{x}) + \mathbf{s} - \mathbf{h}_{\max}) \\ \dot{\mathbf{t}} = k_t \mathbf{I}(\mathbf{h}(\mathbf{x}) - \mathbf{t} - \mathbf{h}_{\min}) \\ \dot{q} = k_q (f(\mathbf{x}) - q) \end{cases} \quad (16)$$

where k_x, k_s, k_t and k_q are proper gain factors.

As we shown in the previous section, the dynamic evolution of the vector $\mathbf{F}(\mathbf{z})$ is governed by the following equation:

$$\dot{\mathbf{F}}(\mathbf{z}(t)) = -k \frac{d\mathbf{F}(\mathbf{z})}{d\mathbf{z}} \left(\frac{d\mathbf{F}(\mathbf{z})}{d\mathbf{z}} \right)^T \mathbf{F}(\mathbf{z}) \quad (17)$$

and, due to the structure of the matrix $\frac{d\mathbf{F}(\mathbf{z})}{d\mathbf{z}} \left(\frac{d\mathbf{F}(\mathbf{z})}{d\mathbf{z}} \right)^T$, we can conclude that $\mathbf{F}(\mathbf{z})$ asymptotically converges to the equilibrium point $\mathbf{F}(\mathbf{z}^*) = 0$.

Therefore, at the equilibrium point, it results:

$$\mathbf{F}(\mathbf{z}^*)^T \mathbf{F}(\mathbf{z}^*) = 0 \quad (18)$$

which can be expanded as follow:

$$\begin{aligned} (f(\mathbf{x}^*) - q^*)^2 &+ \sum_{j=1}^n (g_j(\mathbf{x}^*))^2 \\ &+ \sum_{k=1}^m (h_k(\mathbf{x}^*, \mathbf{s}^*) + s_i - h_{k.\max})^2 \\ &+ \sum_{k=1}^m (h_k(\mathbf{x}^*, \mathbf{t}^*) - t_i - h_{k.\min})^2 = 0 \end{aligned} \quad (19)$$

Since all the terms of this summation are quadratic, we can conclude that \mathbf{x}^* is certainly a feasible solution for the problem (10) (since all the equality and inequality constraints are satisfied).

Now let's analyze the value assumed by the slack variable q at the equilibrium point (namely q^*).

When the dynamic system (16) reaches the equilibrium, it results $\dot{\mathbf{z}} = 0$ (namely $\dot{\mathbf{x}} = \dot{q} = \dot{\mathbf{s}} = \dot{\mathbf{t}} = 0$) and the following set of equations must be satisfied:

$$\begin{cases} \left(\frac{df(\mathbf{x})}{d\mathbf{x}} \Big|_{\mathbf{x}=\mathbf{x}^*} \right)^T (f(\mathbf{x}^*) - q^*) = 0 \\ f(\mathbf{x}^*) - q^* = 0 \\ \left(\frac{d\mathbf{h}(\mathbf{x}^*)}{d\mathbf{x}} \right)^T (\mathbf{h}(\mathbf{x}^*) + \mathbf{s}^* - \mathbf{h}_{\max}) = 0 \\ \mathbf{h}(\mathbf{x}^*) + \mathbf{s}^* - \mathbf{h}_{\max} = 0 \\ \left(\frac{d\mathbf{h}(\mathbf{x}^*)}{d\mathbf{x}} \right)^T (\mathbf{h}(\mathbf{x}^*) - \mathbf{t}^* - \mathbf{h}_{\min}) = 0 \\ \mathbf{h}(\mathbf{x}^*) - \mathbf{t}^* - \mathbf{h}_{\min} = 0 \\ \left(\frac{d\mathbf{g}(\mathbf{x}^*)}{d\mathbf{x}} \right)^T \mathbf{g}(\mathbf{x}^*) = 0 \\ \mathbf{g}(\mathbf{x}^*) = 0 \end{cases} \quad (20)$$

From the first equation it follows that:

$$\left(\frac{df(\mathbf{x})}{d\mathbf{x}} \Big|_{\mathbf{x}=\mathbf{x}^*} \right)^T = 0 \quad (21)$$

This is the first order condition for a minimum of $f(\mathbf{x})$ in the equilibrium point \mathbf{x}^* .

Besides, from the second of (20) it follows that $q(t)$ converge to the value assumed by $f(\mathbf{x})$ in its minimum.

To better clarify this concept let's observe that:

1. the dynamic of $f(\mathbf{x}(t))$ is independent from $q(t)$ [as it can be inferred by (17)];
2. the variable $q(t)$ is forced to track the "input signal" $f(\mathbf{x}(t))$ with an exponential convergence governed by the constant k_q [as it can be inferred by (16)].

Consequently, if we let $q(t)$ to vary slower compared to $f(\mathbf{x}(t))$ (i.e. by reducing k_q) then we expect that equation (21) is satisfied since $\mathbf{x}(t)$ reaches the equilibrium point \mathbf{x}^* quickly compared to $q(t)$.⁵

Therefore, thanks to the introduction and the proper management of the slack variables, the proposed computing paradigm can be adopted to solve scalar constrained optimization problems. In this case, in addition to the previously described benefits, the application of the dynamic paradigm allows us to effectively manage the inequality constraints on the slack and the control/decision variables (namely $s_i \geq 0, t_i \geq 0 \ i = 1, \dots, m$, and $x_{\min,i} \leq x_i \leq x_{\max,i} \ i = 1, \dots, n_x$) by considering proper block saturations without the need for integrating these constraints in equations (14). This feature could be very useful in solving large scale problems.

2.3 Multi-objective programming problems

Let's now consider the following constrained multiobjective programming problem:

$$\begin{cases} \min_{\mathbf{x}} \mathbf{f}(\mathbf{x}) \\ s.t. \ \mathbf{g}(\mathbf{x}) = 0 \\ \mathbf{h}_{\min} \leq \mathbf{h}(\mathbf{x}) \leq \mathbf{h}_{\max} \end{cases} \quad (22)$$

where $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_p(\mathbf{x})]^T$ is the p -dimensional objective function vector.

By following the same approach described in Sect. 3.2, it easy to show as the solution of the problem (22) can be addressed by solving the following nonlinear system of equations:

⁵ During this evolution it results $q(t) \approx q(0) \neq f(\mathbf{x}^*)$. Successively, when $q(t)$ reaches the equilibrium point it results $q^* = f(\mathbf{x}^*)$ and we can conclude that $q(t)$ converges to the value assumed by $f(\mathbf{x})$ in its minimum.

$$\begin{cases} \mathbf{F}_1(\mathbf{x}, \mathbf{q}) = \mathbf{f}(\mathbf{x}) - \mathbf{q} = 0 \\ \mathbf{F}_2(\mathbf{x}) = 0 \\ \mathbf{F}_3(\mathbf{x}, \mathbf{s}) = 0 \\ \mathbf{F}_4(\mathbf{x}, \mathbf{t}) = 0 \end{cases} \quad (23)$$

where \mathbf{q} is the vector of the p slack variables q_i .

The solution of this problem by the proposed computing paradigm requires the time simulation of the following dynamic system:

$$\dot{\mathbf{z}} = -k \left(\frac{d\mathbf{F}(\mathbf{z})}{d\mathbf{z}} \right)^T \mathbf{F}(\mathbf{z}) \quad (24)$$

where $\mathbf{z} = [\mathbf{x}, \mathbf{q}, \mathbf{s}, \mathbf{t}]$ and $\mathbf{F}(\mathbf{z}) = [\mathbf{F}_1(\mathbf{x}, \mathbf{q}), \mathbf{F}_2(\mathbf{x}), \mathbf{F}_3(\mathbf{x}, \mathbf{s}), \mathbf{F}_4(\mathbf{x}, \mathbf{t})]$.

Equation (24) can be expanded as follow:

$$\begin{cases} \dot{\mathbf{x}} = -k_x \left[\sum_{i=1}^p \left(\frac{df_i(\mathbf{x})}{d\mathbf{x}} \right)^T (f_i(\mathbf{x}) - q_i) + \left(\frac{d\mathbf{g}(\mathbf{x})}{d\mathbf{x}} \right)^T \mathbf{g}(\mathbf{x}) \right. \\ \left. + \left(\frac{d\mathbf{h}(\mathbf{x})}{d\mathbf{x}} \right)^T (\mathbf{h}(\mathbf{x}) + \mathbf{s} + \mathbf{h}_{\max}) + \left(\frac{d\mathbf{h}(\mathbf{x})}{d\mathbf{x}} \right)^T (\mathbf{h}(\mathbf{x}) - \mathbf{t} - \mathbf{h}_{\min}) \right] \\ \dot{\mathbf{s}} = -k\mathbf{I}(\mathbf{h}(\mathbf{x}) + \mathbf{s} - \mathbf{h}_{\max}) \\ \dot{\mathbf{t}} = +k\mathbf{I}(\mathbf{h}(\mathbf{x}) - \mathbf{t} - \mathbf{h}_{\min}) \\ \dot{q} = k_q(f(\mathbf{x}) - q) \end{cases} \quad (25)$$

As we have shown in the previous section, at the equilibrium point, it results:

$$\mathbf{F}(\mathbf{z}^*)^T \mathbf{F}(\mathbf{z}^*) = 0 \quad (26)$$

Or equivalently:

$$\begin{aligned} & \sum_{i=1}^p (f_i(\mathbf{x}^*) - q_i^*)^2 + \sum_{j=1}^n (g_j(\mathbf{x}^*))^2 + \sum_{k=1}^m (h_k(\mathbf{x}^*, \mathbf{s}^*) \\ & + s_i - h_{k.\max})^2 + \sum_{k=1}^m (h_k(\mathbf{x}^*, \mathbf{t}^*) - t_i - h_{k.\min})^2 = 0 \end{aligned} \quad (27)$$

From this equation it is worth observing as \mathbf{x}^* is certainly a feasible solution of the problem (22).

Now let's analyze the important role played by the variables q_i .

When the dynamic system (25) reaches the equilibrium it results $\dot{\mathbf{z}} = 0$ (namely $\dot{\mathbf{x}} = \dot{q} = \dot{\mathbf{s}} = \dot{\mathbf{t}} = 0$) and the following set of equations must be satisfied:

$$\begin{cases} \left(\frac{d\mathbf{f}(\mathbf{x}^*)}{d\mathbf{x}} \right)^T (\mathbf{f}(\mathbf{x}^*) - \mathbf{q}^*) = 0 \\ \mathbf{f}(\mathbf{x}^*) - \mathbf{q}^* = 0 \\ \left(\frac{d\mathbf{h}(\mathbf{x}^*)}{d\mathbf{x}} \right)^T (\mathbf{h}(\mathbf{x}^*) + \mathbf{s}^* - \mathbf{h}_{\max}) = 0 \\ \mathbf{h}(\mathbf{x}^*) + \mathbf{s}^* - \mathbf{h}_{\max} = 0 \\ \left(\frac{d\mathbf{h}(\mathbf{x}^*)}{d\mathbf{x}} \right)^T (\mathbf{h}(\mathbf{x}^*) - \mathbf{t}^* - \mathbf{h}_{\min}) = 0 \\ \mathbf{h}(\mathbf{x}^*) - \mathbf{t}^* - \mathbf{h}_{\min} = 0 \\ \left(\frac{d\mathbf{g}(\mathbf{x}^*)}{d\mathbf{x}} \right)^T \mathbf{g}(\mathbf{x}^*) = 0 \\ \mathbf{g}(\mathbf{x}^*) = 0 \end{cases} \quad (28)$$

The first of (28) implies that:

$$\begin{cases} \left[\sum_{i=1}^p \frac{df_i(\mathbf{x})}{dx_1} \Big|_{\mathbf{x}=\mathbf{x}^*} (f_i(\mathbf{x}^*) - q_i^*) \right] = 0 \\ \dots \\ \left[\sum_{i=1}^p \frac{df_i(\mathbf{x})}{dx_n} \Big|_{\mathbf{x}=\mathbf{x}^*} (f_i(\mathbf{x}^*) - q_i^*) \right] = 0 \end{cases} \quad (29)$$

These equations represent the necessary conditions for the local minimization of the squared sum of the objective functions.

This statement could be confirmed by observing that:

- the dynamic of each objective function $f_i(\mathbf{x}(t))$ is not influenced by the dynamic of the corresponding slack variable $q_i(t)$;
- each slack variable $q_i(t)$ is forced to track the "input signal" $f_i(\mathbf{x}(t))$ by an exponential convergence governed by the constant k_q ;

Then if we let $q_i(t)$ to vary slower compared to $f_i(\mathbf{x}(t))$ (i.e. by reducing k_q) and if we assume $q_i(0) = 0$ then we expect that $\mathbf{x}(t)$ reaches an equilibrium point which locally minimize $\sum_{i=1}^p f_i^2(\mathbf{x})$.⁶

Moreover, from the second of (28) it follows that each $q_i(t)$ converges to the value assumed by the i -th objective function at this equilibrium point.

This important result allows us to conclude that the proposed computing paradigm implements a quadratic scalarization technique for constrained multiobjective minimization problems solution.⁷

3 Experimental results

This section discusses the application of the proposed computing paradigm in the task of solving several benchmark problems. In details, we consider non linear systems of equations, scalar and multi-objective optimization problems. In all the experiments discussed in this section we assumed random initial conditions for the dynamical models.⁸ The obtained results are detailed presented in the next subsections.

⁶ Under these hypothesis when $\mathbf{x}(t)$ converges to \mathbf{x}^* it results $q_i(t) \approx q_i(0) = 0 \neq f_i(\mathbf{x}^*)$.

⁷ Intensive research activities aiming at formally defining the connections between the proposed dynamic paradigm, the gradient descent algorithm and the Pareto theory is currently under investigation by the authors.

⁸ We expect that a more accurate selection of the initial states (i.e. by considering a feasible solution generated by a traditional algorithm) could sensibly improve the algorithm convergence especially in solving non convex optimization problems.

3.1 Solving non linear systems of equations

To assess the effectiveness of the proposed computing paradigm in solving non linear systems of equations, we considered the two benchmarks defined in Grosan and Abraham (2008).

The first problem asks for the solution of the following equations:

$$\begin{cases} g_1(x_1, x_2) = \cos(2x_1) - \cos(2x_2) - 0.4 = 0 \\ g_2(x_1, x_2) = 2(x_2 - x_1) + \sin(2x_2) - 2 \sin(x_1) - 1.2 = 0 \end{cases} \quad (30)$$

To solve this problem the application of the proposed computing paradigm requires the time simulation of the following dynamic system:

$$\begin{cases} \dot{x}_1 = -k \left(\frac{dg_1(x_1, x_2)}{dx_1} g_1(x_1, x_2) + \frac{dg_2(x_1, x_2)}{dx_1} g_2(x_1, x_2) \right) \\ \dot{x}_2 = -k \left(\frac{dg_1(x_1, x_2)}{dx_2} g_1(x_1, x_2) + \frac{dg_2(x_1, x_2)}{dx_2} g_2(x_1, x_2) \right) \end{cases} \quad (31)$$

The obtained trajectories are depicted in Fig. 1. Analysing this figure it is worth noting as the system converges to the following state (which represents the problem solution):

$$\begin{cases} x_1^* = 0.1552 \\ x_2^* = 0.4929 \end{cases} \quad (32)$$

In order to assess the benefits deriving by the application of the proposed approach, in Table 1 the solutions obtained by applying other settled solution strategies are summarized (Grosan and Abraham 2008).

By analysing these data it is possible to appreciate the good degree of accuracy characterising the solution identified by the proposed paradigm.

The second problem considered in our studies asks for solving the following equations:

$$\begin{cases} g_1(x_1, x_2) = e^{x_1} + x_1 x_2 - 1 = 0 \\ g_2(x_1, x_2) = \sin(x_1 x_2) + x_1 + x_2 - 1 = 0 \end{cases} \quad (33)$$

By applying the dynamic paradigm we obtained the trajectories depicted in Fig. 2 which converge to the following state:

$$\begin{cases} x_1^* = 0.0004 \\ x_2^* = 0.9991 \end{cases} \quad (34)$$

This solution has been compared with those obtained by applying other settled solution strategies. The corresponding results are summarized in Table 2.

The analysis of these data confirms the effectiveness of the dynamic paradigm in the task of solving non linear systems of equations.

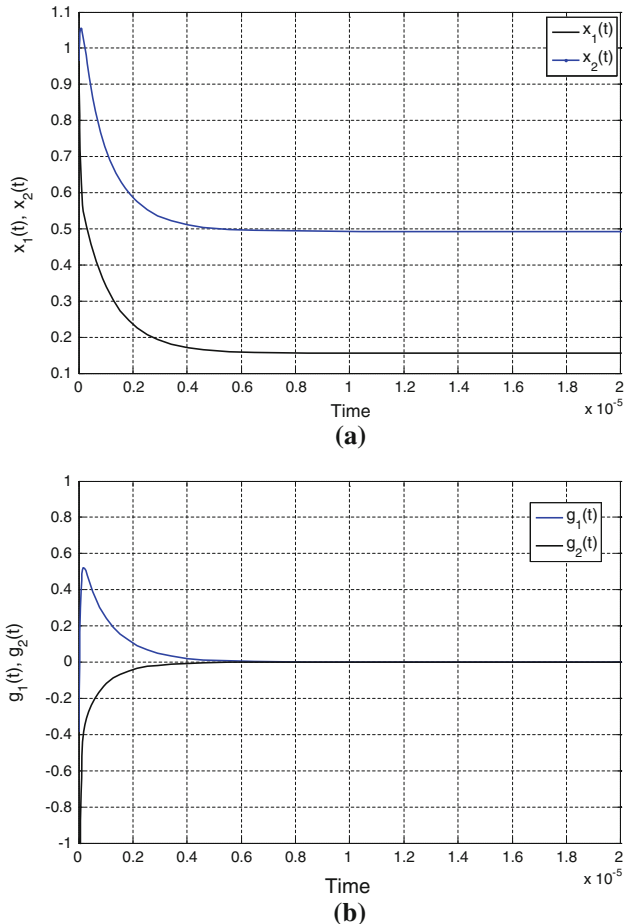


Fig. 1 Solution of the non linear equations system (30): **a** Time evolution of $x_1(t)$ and $x_2(t)$. **b** Time evolution of $g_1(x_1(t), x_2(t))$ and $g_2(x_1(t), x_2(t))$

Table 1 Solving non linear systems of equations: results for the first example

Method	Solution	Function values
Newton’s method	(0.15, 0.49)	(−0.00168, 0.01497)
Secant method	(0.15, 0.49)	(−0.00168, 0.01497)
Broyden’s method	(0.15, 0.49)	(−0.00168, 0.01497)
Effati’s method	(0.1575, 0.4970)	(0.005455, 0.00739)
Evolutionary approach	(0.15772, 0.49458)	(0.001264, 0.000969)
<i>Proposed method</i>	<i>(0.1552, 0.4929)</i>	<i>(−0.6172E−5, −0.3621E−5)</i>

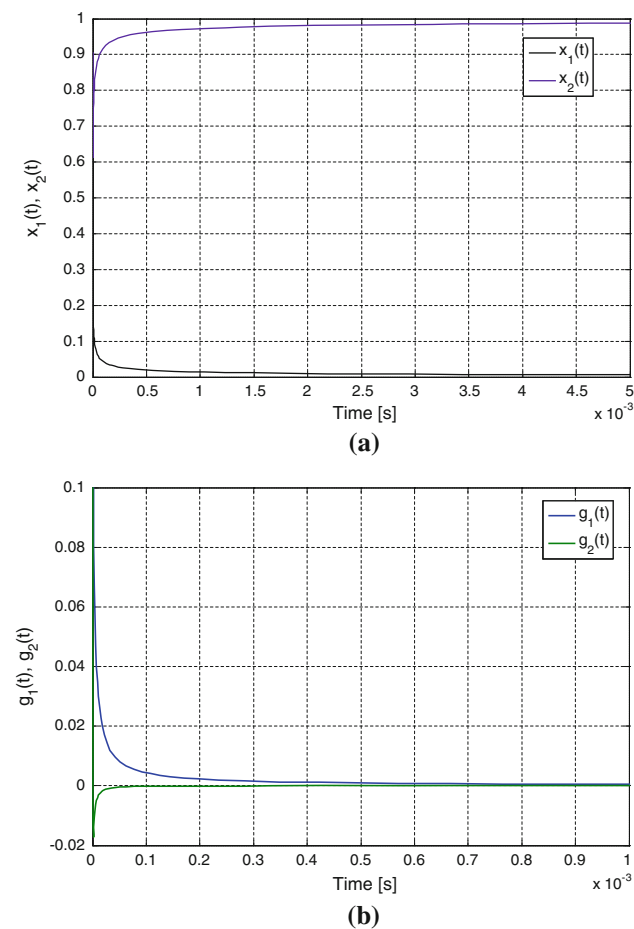


Fig. 2 Solution of the non linear equations system (34): **a** Time evolution of $x_1(t)$ and $x_2(t)$. **b** Time evolution of $g_1(x_1(t), x_2(t))$ and $g_2(x_1(t), x_2(t))$

3.2 Scalar optimization

As stated in Sect. 2.2, the dynamic paradigm can be easily reconfigured in order to address scalar optimization problems. To clarify this concept let's consider the following example:

$$\begin{cases} \min_{x_1, x_2} f(x_1, x_2) = \left(\frac{6x_1}{2+x_1^2+x_2^2} \right) \\ -5 \leq x_1 \leq 5 \\ -5 \leq x_2 \leq 5 \end{cases} \quad (35)$$

This function has a minimum in $(x_1, x_2) = (-\sqrt{2}, 0)$ and the value of this minimum is $f(x_1, x_2) = -2.1213$.

Table 2 Solving non linear systems of equations: results for the second example

Method	Solution	Function values
Effati's method	(0.0096, 0.9976)	(0.019223, 0.016776)
Evolutionary approach	(-0.00138, 1.0027)	(-0.00276, -6.37 E -5)
Proposed method	(0.0004, 0.9991)	(0.4966E-5, 0.0001E-6)

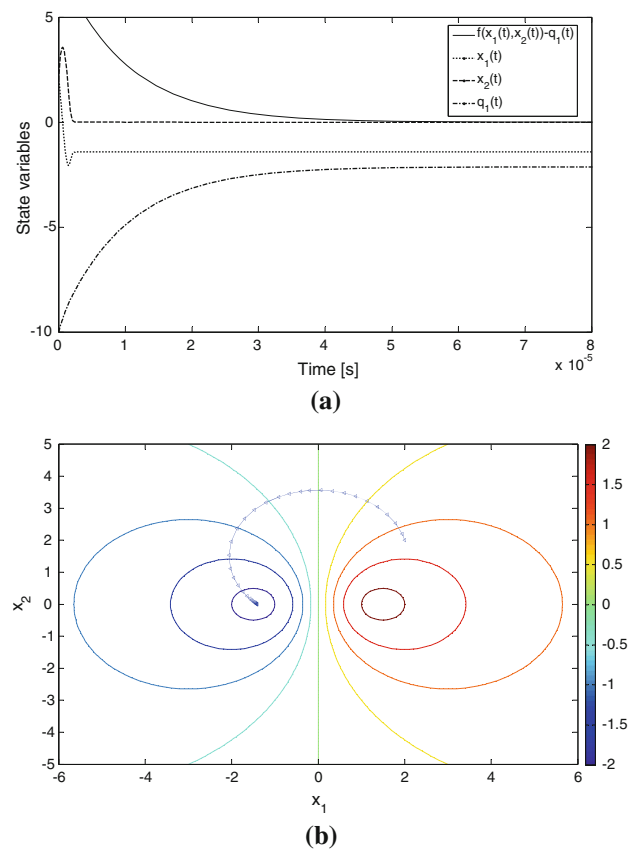


Fig. 3 State variables trajectories—example of scalar minimization. **a** Time evolution of $x_1(t)$, $x_2(t)$, $q_1(t)$ and $f(x_1(t), x_2(t)) - q_1(t)$. **b** Trajectory $(x_1(t), x_2(t))$ and contour plot of $f(x_1, x_2)$

To solve the problem (36), the application of the proposed computing paradigm requires the time simulation of the following dynamic system:

$$\begin{cases} \dot{x}_1 = -k_x \left[\left(\frac{6}{2+x_1^2+x_2^2} - \frac{12x_1^2}{(2+x_1^2+x_2^2)^2} \right) \left(\frac{6x_1}{2+x_1^2+x_2^2} - q_1 \right) \right] \\ \dot{x}_2 = -k_x \left[-\frac{12x_1x_2}{(2+x_1^2+x_2^2)^2} \left(\frac{6x_1}{2+x_1^2+x_2^2} - q_1 \right) \right] \\ \dot{q}_1 = k_q \left(\frac{6x_1}{2+x_1^2+x_2^2} - q_1 \right) \end{cases} \quad (36)$$

Two block saturations aimed at satisfying the inequality constraints on x_1 and x_2 have been integrated in the simulation process.

The obtained trajectories are depicted in Figs. 3, 4. Analysing these figures it is worth noting as the system con-

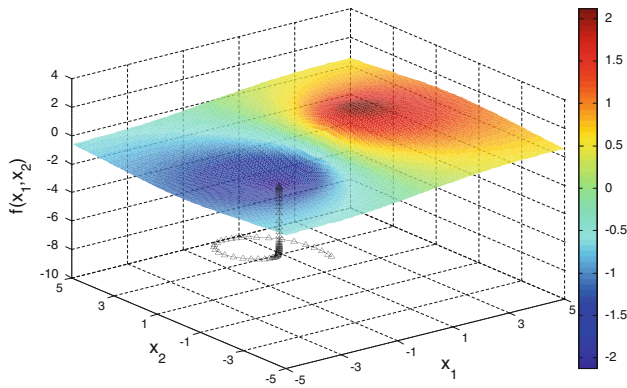


Fig. 4 Trajectory $(x_1(t), x_2(t), q_1(t))$ and 3D plot of $f(x_1, x_2)$ (as expected $q_1(t)$ converge to the function minimum)

verges to the following state:

$$\begin{cases} x_1^* = -\sqrt{2} \\ x_2^* = 0 \\ q_1^* = 2.1213 \end{cases} \quad (37)$$

and, as expected, $x_1(t)$ and $x_2(t)$ converge to the problem solution while $q_1(t)$ converges to the function minimum.

3.3 Multi-objective optimization

The introduction of proper slack variables allows the dynamic paradigm to effectively solve Multi-objective optimization problems. To clarify this concept let's consider the following example:

$$\begin{cases} \min_{x_1, x_2} (f_1(x_1, x_2), f_2(x_1, x_2)) \\ = \left(\frac{10x_1+2}{2+x_1+x_2}, \frac{1}{3} \log \left(\frac{1}{4}x_1^4 + x_2^2 + 4x_1^2 + 4 \right) + 2 \right) \\ -5 \leq x_1 \leq 5 \\ -5 \leq x_2 \leq 5 \end{cases} \quad (38)$$

The solution of this problem by the proposed computing paradigm asks for the time simulation of the following dynamic system:

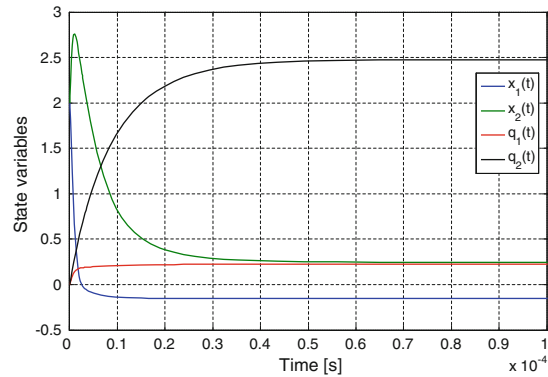
$$\begin{cases} \dot{x}_1 = -k_x \left[\left(\frac{\partial f_1}{\partial x_1} \right) (f_1 - q_1) + \left(\frac{\partial f_2}{\partial x_1} \right) (f_2 - q_2) \right] \\ \dot{x}_2 = -k_x \left[\left(\frac{\partial f_1}{\partial x_2} \right) (f_1 - q_1) + \left(\frac{\partial f_2}{\partial x_2} \right) (f_2 - q_2) \right] \\ \dot{q}_1 = k_q (f_1 - q_1) \\ \dot{q}_2 = k_q (f_2 - q_2) \end{cases} \quad (39)$$

Two block saturations aimed at satisfying the inequality constraints on x_1 and x_2 have been integrated in the simulation process.

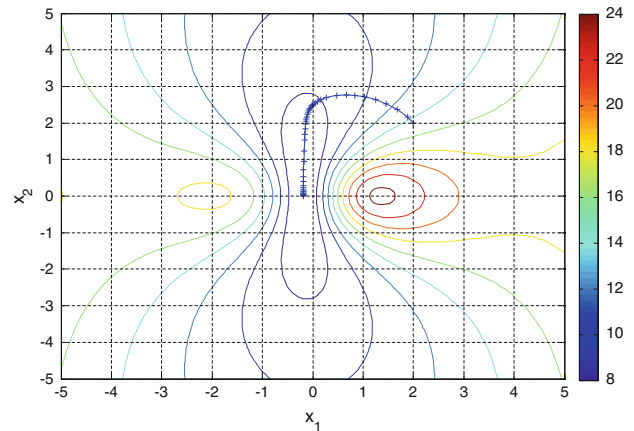
The obtained results are summarized in Table 3 and in Fig. 5, 6. In particular, by analysing Fig. 5 it is possible to observe as $(x_1(t), x_2(t))$ converge to the minimum of the scalar function $f_1^2(x_1, x_2) + f_2^2(x_1, x_2)$. Moreover by analysing Fig. 6,

Table 3 Solution to problem (38)

x_1^*	x_2^*	$q_1^* = f_1(x_1^*, x_2^*)$	$q_2^* = f_2(x_1^*, x_2^*)$
-0.1530	0.2459	0.2257	2.4747



(a)



(b)

Fig. 5 State variables trajectories—example of multi-objective optimization. **a** Time evolution of $x_1(t), x_2(t), q_1(t)$ and $q_2(t)$. **b** Trajectory $(x_1(t), x_2(t))$ and contour plot of $f_1^2(x_1, x_2) + f_2^2(x_1, x_2)$

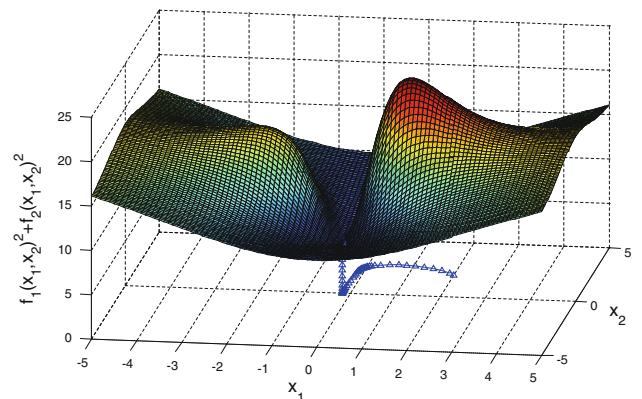


Fig. 6 Trajectory $(x_1(t), x_2(t), q_1^2(t) + q_2^2(t))$ and 3D plot of $f_1^2(x_1, x_2) + f_2^2(x_1, x_2)$

Table 4 Solution to the multi-objective problems (40)–(44)

	\mathbf{x}^*	$\mathbf{q}^* = (f_1(\mathbf{x}^*), f_2(\mathbf{x}^*))$	$\mathbf{s}^* = \mathbf{h}_{\max} - \mathbf{h}(\mathbf{x}^*)$	$\mathbf{t}^* = \mathbf{h}(\mathbf{x}^*) - \mathbf{h}_{\min}$
Problem (40)	(0.6667, 0)	(0.6667, 1.4999)		(0.0005, 5.0005)
Problem (41)	(1.3444, 1.5024)	(16.2592, 25.5962)	9.3794	56.8687
Problem (42)	(−0.1744, 3.2752)	(11.9047, 3.6065)	214.2427	0
Problem (43)	(−3.1416, 0.5912)	(2.2926, 2.5518)		
Problem (44)	(0.000, 0.000, 0.000)	(0.6321, 0.6321)		

it is possible to note as the trajectory of $q_1^2(t) + q_2^2(t)$ converges to the value assumed by $f_1^2(x_1, x_2) + f_2^2(x_1, x_2)$ in its minimum.

Finally, in order to assess the effectiveness of the dynamic computing paradigm in identifying efficient solutions, the following benchmarks have been solved and the obtained solutions have been compared to the corresponding Pareto optimal fronts. The latter have been generated by applying an evolutionary based multi-objective optimization algorithm Zitzler and Thiele (1998).

$$\begin{cases} \min_{(x_1, x_2)} \begin{pmatrix} f_1 = x_1 \\ f_2 = \frac{1+x_2}{x_1} \end{pmatrix} \\ s.t. \\ x_2 + 9x_1 \geq 6 \\ -x_2 + 9x_1 \geq 1 \\ 0.1 \leq x_1 \leq 1 \\ 0 \leq x_2 \leq 5 \end{cases} \quad (40)$$

$$\begin{cases} \min_{(x_1, x_2)} \begin{pmatrix} f_1 = 4x_1^2 + 4x_2^2 \\ f_2 = (x_1 - 5)^2 + (x_2 - 5)^2 \end{pmatrix} \\ s.t. \\ (x_1 - 5)^2 + x_2^2 \leq 25 \\ (x_1 - 8)^2 + (x_2 + 3)^2 \geq 7.7 \\ 0 \leq x_1 \leq 5 \\ 0 \leq x_2 \leq 3 \end{cases} \quad (41)$$

$$\begin{cases} \min_{(x_1, x_2)} \begin{pmatrix} f_1 = (x_1 - 2)^2 + (x_2 - 1)^2 + 2 \\ f_2 = (9x_1 + (x_2 - 1)^2 \end{pmatrix} \\ s.t. \\ x_1^2 + x_2^2 \leq 225 \\ -x_1 + 3x_2 \geq 10 \\ -20 \leq x_1 \leq 20 \\ -20 \leq x_2 \leq 20 \end{cases} \quad (42)$$

$$\begin{cases} \min_{(x_1, x_2)} \begin{pmatrix} f_1 = 1 + (k_1 - 0.5 \sin(x_1) - 2 \cos(x_1) \\ \quad + \sin(x_2) - 1.5 \cos(x_2))^2 \\ \quad + (k_2 - 1.5 \sin(x_1) - \cos(x_1) \\ \quad + 2 \sin(x_2) - 0.5 \cos(x_2))^2 \\ f_2 = (x_1 + 3)^2 + (x_2 + 1)^2 \end{pmatrix} \\ s.t. \\ -\pi \leq x_1 \leq \pi \\ -\pi \leq x_2 \leq \pi \end{cases} \quad (43)$$

$$\begin{cases} \min_{(x_1, x_2, x_3)} \begin{pmatrix} f_1 = 1 - e^{-\sum_{i=1}^3 (x_i - \frac{1}{\sqrt{3}})^2} \\ f_2 = 1 - e^{-\sum_{i=1}^3 (x_i + \frac{1}{\sqrt{3}})^2} \end{pmatrix} \\ s.t. \\ -4 \leq x_1 \leq 4 \\ -4 \leq x_2 \leq 4 \\ -4 \leq x_3 \leq 4 \end{cases} \quad (44)$$

The obtained results have been summarized in Table 4 and in Fig. 7.

Analyzing these data it is worth observing as in all cases the dynamic paradigm identifies an efficient solution in the Pareto sense.

4 Toward a reconfigurable hardware architecture

The future direction of our research will be oriented in deploying the proposed computing paradigm according to the reconfigurable hardware architecture schematically depicted in Fig. 8. The insight is to implement the stable artificial model by a proper combination of discrete electronic devices (i.e. operational amplifiers). In this case the variable t is no longer an artificial parameter but it represents the real computing time.

An important feature characterising this architecture is the intrinsic data filtering capability derived by the integral action of the computational process. This make the proposed approach particularly useful in solving programming problems in the presence of uncertainty or noisy data.

Finally, we expect that the hardware deployment of the proposed paradigm will sensibly reduce the convergence times making it an ideal candidate for solving complex and large scale programming problems in near real time.

5 Conclusions

In this paper a unified computing paradigm aimed at solving mathematical programming problems has been proposed. In particular we demonstrated that a generic programming problem can be formalized by a proper set of ordinary differential equations, whose equilibrium points correspond to the

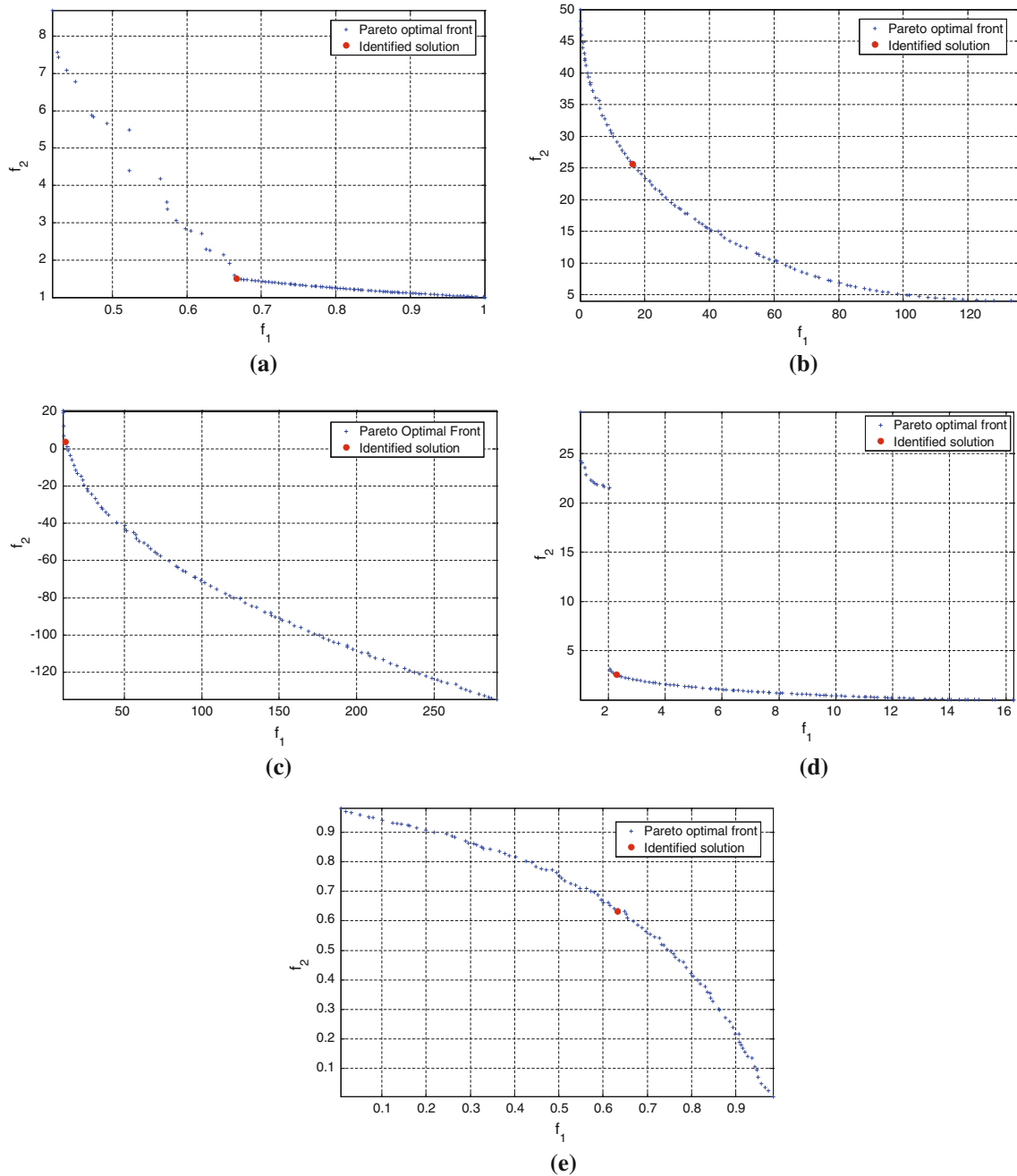


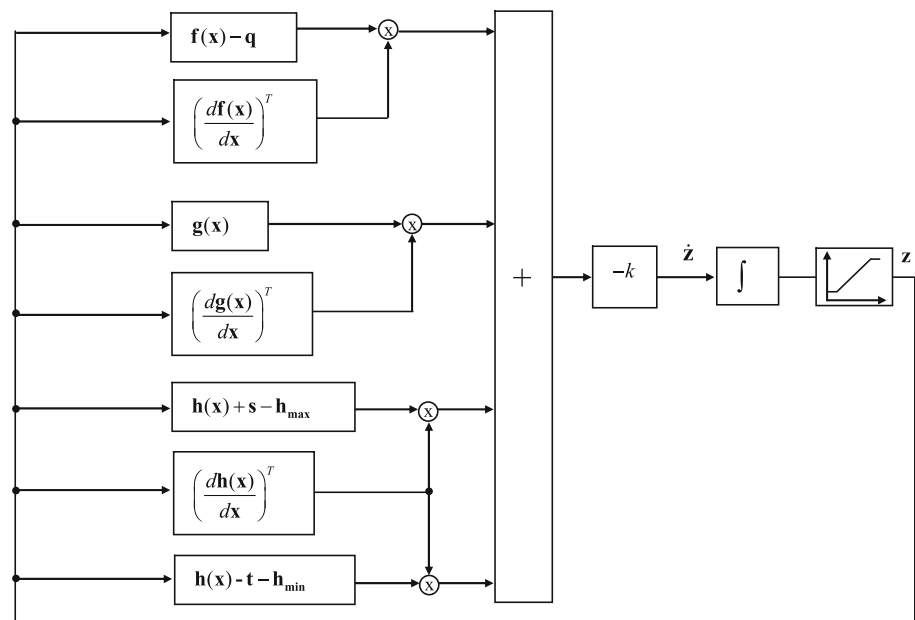
Fig. 7 Solutions to standard multi-objective programming problems: **a** Problem (40). **b** Problem (41). **c** Problem (42). **d** Problem (43). **e** Problem (44)

problem solutions. Besides we rigorously demonstrated that this artificial dynamic system could be explicitly designed to be stable with an exponential asymptotic convergence to an equilibrium point. This equilibrium point represents a root of a system of equations (when applied to non linear systems solution), a feasible solution which locally minimizes the objective function (when applied to scalar optimization problems) or a feasible Pareto solution (when applied to multi objective optimization problems). Thanks to these important

features the generic programming problem can be solved by the time simulation of a stable dynamic system without the need for any kind of matrixes inversion/manipulation.

Extensive numerical results obtained on several benchmarks confirm the intrinsic benefits deriving by the application of the proposed computing paradigm. In particular we observed as it effectively solved systems of non linear equations and converged to feasible local optima and Pareto optimal solutions when applied in solving scalar and multi

Fig. 8 Schematic diagram of the computational process



objective optimization problems respectively. Besides, in all these numerical experiments we observed as the artificial dynamic system exponentially converged to a stable equilibrium point as rigorously demonstrated by the theoretical analysis presented in Sect. 2.

Finally we would like to outline that the application of the dynamic computing paradigm for large scale problems could pose some computational difficulties. In addressing this flaw, it is possible to exploit the intrinsic parallelism characterizing the proposed algorithm. In this connection the authors developed and tested a powerful parallelization algorithm aimed at effectively deploying the proposed dynamic architecture on a distributed computing environment. Besides the authors are improving the theoretical framework presented in this paper in order to address integer and mixed integer programming problems.

Due to space limitation, the theoretical background and the obtained experimental results will be presented in a separate paper.

References

- Agrawal S, Dashora Y, Tiwari MK, Young-Jun S (2008) Interactive particle swarm: a pareto-adaptive metaheuristic to multiobjective optimization. *IEEE Trans Syst Man Cybern Part A Syst Hum* 38(2):258–277
- Astfalk G, Lustig I, Marsten R, Shanno D (1992) The interior-point method for linear programming. *IEEE Softw* 9(4):61–68
- Boggs PT, Domich PD, Rogers JE (1996) An interior-point method for general large scale quadratic programming problems. *Ann Oper Res* 62(1):419–437
- Broyden CG (1965) A class of methods for solving nonlinear simultaneous equations. *Math Comput* 19(92):577–593
- Chinchuluun A, Pardalos PM (2007) A survey of recent developments in multiobjective optimization. *Ann Oper Res* 154:29–50
- Coello CA, Aguirre AH, Zitzler E (2007) Evolutionary multi-objective optimization. *Eur J Oper Res* 181(3):1617–1619
- Conn AR, Gould NIM, Toint PL (2000) *Trust-Region Methods*. SIAM, Philadelphia
- Delbos F, Gilbert JC (2005) Global linear convergence of an augmented Lagrangian algorithm for solving convex quadratic optimization problems. *J Convex Anal* 12(1):45–69
- Denis JE, Wolkowicz H (1993) Least change secant methods, sizing, and shifting. *SIAM J Numer Anal* 30:1291–1314
- Fonseca CM, Fleming PJ (1998) Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation. *IEEE Trans Syst Man Cybern Part A Syst Hum* 28(1):26–37
- Grosan C, Abraham A (2008) A new approach for solving nonlinear equations systems. *IEEE Trans Syst Man Cybern Part A Syst Hum* 38(3):698–714
- Li Y-X, Gen M (1996) Nonlinear mixed integer programming problems using genetic algorithm and penalty function. *IEEE Int Conf Syst Man Cybern* 2677–2682(4):14–17
- Malick J, Roupin F (2013) On the bridge between combinatorial optimization and nonlinear optimization: a family of semidefinite bounds for 0–1 quadratic problems leading to quasi-Newton methods. *Math Progr* 140(1):99–124
- Marler RT, Arora JS (2004) Survey of multi-objective optimization methods for engineering. *Struct Multidiscip Optim* 26(6):369–395
- Martinez JM (1994) Algorithms for solving nonlinear systems of equations. In: Spedicato E (ed) *Continuous optimization: the state of the art*. Kluwer, Norwell, pp 81–108
- Michalewicz Z (1995) Genetic algorithms, numerical optimization, and constraints. In: *Proceedings of the 6th international conference on genetic algorithms*, pp 151–158
- Narula SC, Kirilov L, Vassilev V (1994) Reference direction approach for solving multiple objective nonlinear programming problems. *IEEE Trans Syst Man Cybern* 24:859:804–806

- Ng KYK (1987) Goal Programming. Method of weighted residuals, and optimal control problems. *IEEE Trans Syst Man Cybern* 17(1):102–106
- Okabe T, Jin Y, Sendhoff B (2003) A critical survey of performance indices for multi-objective optimisation. In: *The 2003 Congress on Evolutionary Computation, CEC '03*, vol 2, pp 878–885
- Ortega JM, Rheinboldt WC (1970) *Iterative solution of nonlinear equations in several variables*. Academic, New York
- Park KS (2004) Mathematical programming models for characterizing dominance and potential optimality when multicriteria alternative values and weights are simultaneously incomplete. *IEEE Trans Syst Man Cybern Part A Syst Hum* 34(5):601–614
- Rodriguez-Vazquez K, Fonseca CM, Fleming PJ (2004) Identifying the structure of nonlinear dynamic systems using multiobjective genetic programming. *IEEE Trans Syst Man Cybern Part A Syst Hum* 34(4):531–545
- Tinney WF, Walker JW (1967) Direct solutions of sparse network equations by optimally ordered triangular factorization. *Proc IEEE* 55(11):1801–1809
- Torelli F, Vaccaro A, Xie N (2013) A novel optimal power flow formulation based on the Lyapunov Theory. *IEEE Trans Power Syst* 28(4):4405–4415
- Xie N, Torelli F, Bompard E, Vaccaro A (2013) A dynamic computing paradigm for comprehensive power flow analysis. *IET Gener Transm Distribn* 7(8):832–842
- Yang X-S (2011) Metaheuristic optimization: algorithm analysis and open problems. *Experimental Algorithms*. In: *Lecture notes in computer science*, vol 6630, pp 21–32
- Zeshui X, Xiaoqiang C, Shousheng L (2011) Nonlinear programming model integrating different preference structures. *IEEE Trans Syst Man Cybern Part A Syst Hum* 41(1):169–177
- Zitzler E, Thiele L (1998) Multiobjective optimization using evolutionary algorithms: a comparative case study. In: *Parallel problem solving from nature*, Amsterdam, The Netherlands. Springer, Berlin, pp 292, 301