Computers and Chemical Engineering 71 (2014) 154–157

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

On the equivalence between the modifier-adaptation and trust-region frameworks

Gene A. Bunin

39 Blue Jay Lane, Ashland, MA 01721, United States

ARTICLE INFO

Article history:
Received 23 December 2013
Received in revised form 28 July 2014
Accepted 31 July 2014
Available online 8 August 2014

Keywords:
Modifier adaptation
Trust-region methods
Real-time optimization

ABSTRACT

In this short note, the recently popular modifier-adaptation framework for real-time optimization is discussed in tandem with the well-developed trust-region framework of numerical optimization, and it is shown that the basic version of the former is a simplification of the latter when the problem is unconstrained. This relation is then exploited to propose a globally convergent modifier-adaptation algorithm using already developed trust-region theory. Cases when the two may not be equivalent and extensions to constrained problems are also discussed.

1. The real-time optimization problem

In the process systems engineering community, the basic idea of most real-time optimization (RTO) schemes consists in finding a set of optimal operating conditions – often steady-state setpoints in a multilayer hierarchical scheme – that minimize (resp., maximize) the steady-state cost (resp., profit) of some given plant subject to constraints (Brdys and Tatjewski, 2005). While models of the process being optimized are often available, it is generally the case that they are either inaccurate and/or incomplete, which motivates the data-driven "real-time" element of RTO, thereby forcing the optimization algorithm to use the measurements obtained from the process as feedback to modify the provided setpoints so as to ultimately reject the model uncertainty and converge to the optimal conditions of the plant.

A fairly general mathematical formulation of this problem that suffices for many practical cases is as follows:

$$\begin{aligned} &\underset{\mathbf{u}}{\text{minimize}} \quad \phi_p(\mathbf{u}) \\ &\text{subject to} \;\; g_{p,j}(\mathbf{u}) \le 0, \quad j = 1, \ldots, n_g, \end{aligned} \tag{1}$$

where $\mathbf{u} \in \mathbb{R}^{n_u}$ denote the decision variables, or the "inputs", of the problem, while the functions $\phi, g : \mathbb{R}^{n_u} \to \mathbb{R}$ denote the cost and constraints, respectively. The subscript p (for "plant") is used to indicate that the function corresponds to an experimental relationship that is not perfectly known and may only be approximated by a model, which we will mark with the subscript $\hat{p}$ (e.g., $\phi_{\hat{p}}$ being the model approximation of ϕ_p). In the simplest terms, the goal of an RTO algorithm is to solve Problem (1) by iterative experimentation, generating a sequence of steady-state $\mathbf{u}$ values that converges to the plant optimum.

For the majority of this document, we will not focus on Problem (1) but on the unconstrained case

$$\underset{\mathbf{u}}{\text{minimize}} \quad \phi_p(\mathbf{u}), \tag{2}$$

as this is sufficient to convey the main message. We will, however, return to Problem (1) in the end in passing, providing references to works where it is discussed properly and in much greater detail.

2. Review of the modifier-adaptation framework

An approach to solving (1) that has recently gained popularity in the research community is that of *modifier adaptation*, which originally dates back to the work of Roberts (1978) and owes its numerous refinements and fundamental ideas to the *ISOPE* ("iterative setpoint optimization and parameter estimation") framework (Brdys and Tatjewski, 2005). Recent works by Gao and Engell (2005), Chachuat et al. (2009), and Marchetti et al. (2009) have given the approach its modern form by accounting for plant-model mismatch in both the cost and constraints. A number of works in the past few years have also considered various particular aspects of the framework, such as mathematical reformulations to ease or better accommodate particular problem types (François and Bonvin, 2013; Serralunga et al., 2013; Costello et al., 2013), important implementation aspects (Marchetti et al., 2010; Rodger, 2010; Bunin et al., 2012), and major theoretical issues like feasibility

E-mail address: gene.a.bunin@ccapprox.info

http://dx.doi.org/10.1016/j.compchemeng.2014.07.028
0098-1354/© 2014 Elsevier Ltd. All rights reserved.

(Bunin et al., 2011; Navia et al., 2012) and global convergence (Faulwasser and Bonvin, 2014).

The basic philosophy of modifier adaptation lies in applying local corrections to an inherently incorrect model at each RTO iteration k, and solving this corrected version to obtain the following iterate at $k+1$. For the unconstrained case, this would lead to the following update:

$$\mathbf{u}_{k+1} \in \arg\underset{\mathbf{u}}{\text{minimize}} \quad \phi_{\hat{p}}(\mathbf{u}) + \boldsymbol{\lambda}_k^T\mathbf{u}, \tag{3}$$

with the *modifiers* $\boldsymbol{\lambda}_k$, defined as

$$\boldsymbol{\lambda}_k := \nabla\phi_p(\mathbf{u}_k) - \nabla\phi_{\hat{p}}(\mathbf{u}_k), \tag{4}$$

serving to ensure that the plant and corrected model have matching first derivatives at the current iterate $\mathbf{u}_k$.

Placing this into algorithmic form yields the following basic implementation.

Algorithm 1 *(Basic modifier-adaptation algorithm).*

1. *Initialization*: The initial point, $\mathbf{u}_0$, is provided. Set $k := 0$.
2. *Modifier computation*: Compute the modifiers $\boldsymbol{\lambda}_k$ according to (4).
3. *New input calculation*: Obtain $\mathbf{u}_{k+1}$ by solving Problem (3) and apply this set of inputs to the plant.
4. *Iterate*: Set $k := k+1$ and return to Step 2.

The key oft-stated motivation for applying this algorithm is the following upon-convergence guarantee.

Theorem 1. *(First-order critical point upon convergence). Assume that the minimization of (3) always yields a first-order critical point of the modified objective function $\phi_{\hat{p}}(\mathbf{u}) + \boldsymbol{\lambda}_k^T\mathbf{u}$ and that Algorithm 1 has converged to a fixed point $\mathbf{u}_\infty$. It follows that $\mathbf{u}_\infty$ is a first-order critical point of ϕ_p.*

Proof. The result follows immediately from the fact that a first-order critical point for an unconstrained problem is defined entirely by the function's derivatives at that point. As these must match for the modified model and the plant at any iterate, including $\mathbf{u}_\infty$, it follows that finding a first-order critical point for the modified function implies finding one for the plant. □

3. The basic trust-region algorithm

A theoretically rigorous approach for iteratively minimizing a nonlinear function in the mathematical optimization context is that of trust-region methods. In this section, we will consider what attempting to solve Problem (2) in this framework would entail.

Let us start by stating the basic trust-region algorithm for solving (2). This is essentially the algorithm provided in the well-known monograph on trust-region methods (Conn et al., 2000, Ch. 6) but with a few additional simplifications and some notational changes. Namely, we use the 2-norm instead of the general p-norm and explicitly distinguish between the reference iterates, $\mathbf{u}_k^*$, and the iterates applied to the plant, $\mathbf{u}_k$.

Algorithm 2 *(Basic trust-region algorithm).*

1. *Initialization*: The initial point, $\mathbf{u}_0$, and initial trust-region radius, $\Delta_0 > 0$, are provided, together with the constants η_1, η_2, γ_1, and γ_2 satisfying $0 < \eta_1 \le \eta_2 < 1$ and $0 < \gamma_1 \le \gamma_2 < 1$. Set $k := 0$, $\mathbf{u}_0^* := \mathbf{u}_0$, and apply $\mathbf{u}_0$ to the plant to obtain $\phi_p(\mathbf{u}_0^*) = \phi_p(\mathbf{u}_0)$.
2. *Model construction*: Construct the model m_k, which is an approximation of ϕ_p over the trust region $\mathcal{B}(\mathbf{u}_k^*, \Delta_k)$, i.e., over a Euclidean ball of radius Δ_k centered at $\mathbf{u}_k^*$.
3. *New input candidate calculation*: Compute a candidate point $\mathbf{u}_{k+1} \in \mathcal{B}(\mathbf{u}_k^*, \Delta_k)$ that "sufficiently reduces the model" m_k.
4. *Acceptance of the candidate point*: Apply $\mathbf{u}_{k+1}$ to the plant to obtain $\phi_p(\mathbf{u}_{k+1})$. Define:
$$\rho_k := \frac{\phi_p(\mathbf{u}_k^*) - \phi_p(\mathbf{u}_{k+1})}{m_k(\mathbf{u}_k^*) - m_k(\mathbf{u}_{k+1})}. \tag{5}$$
If $\rho_k \ge \eta_1$, then set $\mathbf{u}_{k+1}^* := \mathbf{u}_{k+1}$. Otherwise, set $\mathbf{u}_{k+1}^* := \mathbf{u}_k^*$.
5. *Trust-region radius update*: Set Δ_{k+1} such that
$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty) & \text{if } \rho_k \ge \eta_2, \\ [\gamma_2\Delta_k, \Delta_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1\Delta_k, \gamma_2\Delta_k] & \text{if } \rho_k < \eta_1. \end{cases} \tag{6}$$
6. *Iterate*: Set $k := k+1$ and return to Step 2.

Let us now state the assumptions sufficient to prove the global convergence of Algorithm 2 to a first-order critical point (Conn et al., 2000). The following are assumed about the nature of the plant:

Assumption 1. ϕ_p is $\mathcal{C}^2$ (twice continuously differentiable) on $\mathbb{R}^{n_u}$.

Assumption 2. ϕ_p is lower-bounded on $\mathbb{R}^{n_u}$.

Assumption 3. The Hessian of ϕ_p is upper-bounded on $\mathbb{R}^{n_u}$.

As mentioned in Conn et al. (2000), Assumption 3 is often too strong and could actually be restricted to the subspace of $\mathbb{R}^{n_u}$ where the iterates lie. However, as this subspace is not known *a priori*, $\mathbb{R}^{n_u}$ is used for notational convenience.

The following assumptions are made on the model:

Assumption 4. For all k, m_k is $\mathcal{C}^2$ over $\mathcal{B}(\mathbf{u}_k^*, \Delta_k)$.

Assumption 5. m_k matches ϕ_p locally to first order at every k, i.e.:

$$m_k(\mathbf{u}_k^*) = \phi_p(\mathbf{u}_k^*), \tag{7}$$

$$\nabla m_k(\mathbf{u}_k^*) = \nabla\phi_p(\mathbf{u}_k^*). \tag{8}$$

Finally, one requires the following assumption on the algorithm used to solve the trust-region subproblem with regard to its ability to achieve "sufficient reduction" in the model:

Assumption 6. There exists a constant $\kappa \in (0, 1)$ such that for all k:

$$m_k(\mathbf{u}_k^*) - m_k(\mathbf{u}_{k+1}) \ge \kappa\|\nabla m_k(\mathbf{u}_k^*)\| \min\left[\frac{\|\nabla m_k(\mathbf{u}_k^*)\|}{\beta_k}, \Delta_k\right], \tag{9}$$

with $\beta_k > 1$ a finite constant.

One may then state the following.

Theorem 2 (Global convergence to a first-order critical point). *If Assumptions 1–6 are satisfied, it then follows that the iterates generated by Algorithm 2 converge asymptotically to a first-order critical point, i.e.:*

$$\lim_{k\to\infty} \|\nabla\phi_p(\mathbf{u}_k^*)\| = 0. \tag{10}$$

Proof. The reader is referred to Theorem 6.4.6 in Conn et al. (2000). Note that we have, for simplicity, used a slightly stronger assumption and have assumed that m_k is $\mathcal{C}^2$ over $\mathcal{B}(\mathbf{u}_k^*, \Delta_k)$. The two assumptions made by Conn et al. (2000) – namely, that over $\mathcal{B}(\mathbf{u}_k^*, \Delta_k)$ the model m_k is twice differentiable and that its Hessian is bounded – are implied by the single $\mathcal{C}^2$ assumption here. □

4. Equivalence and a globally convergent modifier-adaptation scheme

Both the modifier-adaptation and trust-region algorithms seek to minimize ϕ_p by iteratively optimizing a local approximation of ϕ_p around each $\mathbf{u}_k^*$. The key differences between the two may be summarized as follows: