Computers and Chemical Engineering 81 (2015) 344–354

Contents lists available at ScienceDirect

Computers and Chemical Engineering

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

Parallel optimization by means of a Spectral-Projected-Gradient approach

Juan Ignacio Ardenghi[b], Gustavo Esteban Vazquez[b,c], Nélida Beatriz Brignole[a,b,*]

[a] Planta Piloto de Ingeniería Química (PLAPIQUI) Complejo CCT-UAT, CONICET, Camino La Carrindanga Km. 7, 8000 Bahía Blanca, Argentina
[b] Laboratorio de Investigación y Desarrollo en Computación Científica (LIDeCC), Departamento de Ciencias e Ingeniería de la Computación (DCIC), Universidad Nacional del Sur (UNS), Av. Alem 1253, Bahía Blanca, Argentina
[c] Universidad Católica del Uruguay, Facultad de Ingeniería y Tecnologías, Av. 8 de Octubre 2801, Montevideo, CP 11600, Uruguay

ARTICLE INFO

Article history:
Received 31 October 2014
Received in revised form 2 April 2015
Accepted 8 April 2015
Available online 17 April 2015

Keywords:
Optimization
Parallel programming
Non-linear problems
Efficiency

ABSTRACT

The judicious exploitation of the inherent optimization capabilities of the Spectral-Projected-Gradient method (SPG) is proposed. SPG was implemented in order to achieve efficiency. The novel adjustments of the standard SPG algorithm showed that the parallel approach proves to be useful for optimization problems related to process systems engineering. Efficiency was achieved without having to relax the problems because the original model solutions were obtained in reasonable time.

1. Introduction

It is widely recognized that parallelizing is far from trivial. Parallel models that run in multiple processors are highly modified versions of the corresponding sequential solvers. In fact, parallel algorithms constitute new techniques with their own advantages and drawbacks. Though they can succeed in being faster, their main design difficulties are sometimes related to communication management, interaction between runs, memory requirements or experimental evaluation (Alba, 2005).

El-Rewini and Lewis (1997) had also pointed out that parallel programming involves all the difficulties that comprehend serial programming, together with additional challenges, such as data or task partitioning, parallel debugging, and synchronization. Unlike single-processors systems, interconnection bandwidth and message latency dominate the performance of parallel systems. Moreover, there is no evident way to predict the performance of a new system. Therefore, prior to a significant investment of time and effort, it is difficult to envisage clearly the benefits of parallelizing.

Parallel programming does involve several daunting challenges, but it is worthwhile! Problems not solved before become nowadays solvable by using parallel algorithms. According to Buzzi-Ferraris and Manenti (2010), parallel computing on personal computers is taking its steps as a silent revolution that directly involves many other scientific and industrial areas, naturally including Process Systems Engineering (PSE) and Computer-Aided Process Engineering (CAPE) communities.

In the search of more realistic formulations, the need for more rigorous modelling, but together with the modern global requirement of faster solutions, has grown. Hence, cost-effective solutions are required in order to be able to address effectively large-scale problems, which have proved to be very demanding in terms of computational effort and efficacy, i.e. the length of time devoted to problem-solving.

These days parallel programming has turned into an attractive field that deserves to be carefully exploited. It is growing fast, with an enormous application potential. Then, it constitutes a promising methodology needing more attention by the PSE optimization community. How to address time-consuming problems is undoubtedly a subject of interest and concern for chemical engineers, who have sometimes resorted to parallelism. For instance, Abdel-Jabbar et al. (1998) implemented a partially decentralized state observer on multicomputers demonstrating the potential of parallel processing in the field of model-based control. In turn, Chen et al. (2011) have also accelerated their molecular weight distribution (MWD)

* Corresponding author at: Planta Piloto de Ingeniería Química (PLAPIQUI) Complejo CCT-UAT, CONICET, Camino La Carrindanga Km. 7, 8000 Bahía Blanca, Argentina. Tel.: +54 291 4861700.
E-mail address: dybrigno@criba.edu.ar (N.B. Brignole).

http://dx.doi.org/10.1016/j.compchemeng.2015.04.010
0098-1354/

calculation method by means of parallel programming. Cheimarios et al. (2013) also exploited parallelism when modelling a chemical vapor deposition (CVD) reactor. The time-consuming computations in the micro-scale were efficiently accelerated thanks to the implementation of a synchronous master-worker parallel technique. In turn, Laird et al. (2011) have addressed large-scale dynamic optimization problems with a decomposition approach helpful to exploit parallel computing for the Karush–Kuhn–Tucker (KKT) system. Lately, for the contingency-constrained alternating current optimal power flow (ACOPF) problem Kang et al. (2014) have reported the achievement of significant improvements in solution times by means of their parallel Schur-complement based, nonlinear interior-point method.

Improving both the convergence and solution time of process system optimization problems is nowadays of great significance. The community would greatly benefit by deepening HPC knowledge across the chemical engineering field, especially research involving industry-standard software development and implementation (Piccione, 2014).

Broadly speaking, for the past two decades many non-linear optimization problems in the PSE area have been solved by posing them with a wide variety of strategies, such as parallel processing, model reformulation, model decompositions, convergence-depth control and surrogate-based approaches. Efficient non-traditional algorithmic alternatives have been proposed in order to reduce the computational cost of solving some demanding problems. Kheawhom (2010) reported a constraint handling scheme that exhibited a considerably lower computational cost than the cost required by the traditional penalty function. In turn, Kraemer et al. (2009) proposed a reformulation for complex large-scale distillation processes that required significantly less computational time in order to identify local optima of better quality. In process design and control, Wang et al. (2007) showed advantageous numerical results by using convergence depth control. For process control, Abdel-Jabbar et al. (1998) designed a parallel algorithm that guaranteed stability and optimal performance of the parallel observer. Later, for the problem of integrated design and control optimization of process plants, Egea et al. (2007) proposed surrogate-based methods that compete with conventional control strategies.

Besides, for optimization problems related to planning issues, and always working from a sequential point of view – i.e. without exploiting any opportunity of parallelism – You et al. (2011) aimed at the reduction of computational time by means of model reformulation. With a view to solving large-scale instances effectively, they proposed the following computational strategies: (I) a two-level solution strategy and (II) a continuous approximation method. Their approaches led to the same optimal solutions, but with different CPU times. Moreover, for their problem about the simultaneous route selection and tank sizing approach, You et al. (2011) pointed out that solving the aggregated model may become intractable as the problem size increases, due to the combinatorial complexity of route enumeration.

In contrast, the PSE problems are sometimes solved by relaxing variables and conditions, thus generating non-linear subproblems easier to tackle through linear and quadratic approximations. An important and interesting question is the following: Can we achieve efficiency without having to reformulate nor to relax the problem?

Nowadays, it seems natural that the trust-region methods occupy a significant place in the PSE simulation area. Nevertheless, Spectral Projected Gradient (SPG) methods constitute an inspiration for the acceleration of optimization algorithms via parallelization. SPG (Birgin et al., 2000) was born from the merging of the Barzilai–Borwein (spectral) non-monotone concepts with classical projected gradient strategies (Bertsekas, 1976). Some authors, like Raydan (1997) and Fletcher (2005), analyzed this kind of methods carefully.

2. Solving optimization problems by taking advantage of parallel processing

We can rely on parallel computing in order to reduce computing times significantly, without being necessary to resort to strategies that imply model simplifications or problem reformulations. It is efficient and practical, though it is a daunting challenge to program it carefully. Moreover, parallel computing is useful to complement other approaches. In this work parallel programming has been applied to enhance an algorithm originally proposed by Birgin et al. (2000), who developed a method that was born as a combination of spectral nonmonotone ideas (Grippo et al., 1986) with classical projected gradient strategies (Barzilai and Borwein, 1988).

2.1. The mathematical problem

The optimization problem considered here is a non-linear programming (NLP) problem. Its model involves a non-linear objective function $f(x)$ subjected to a set of equality constraints $c_i(x) = 0$, $i = 1, \ldots, n_i$; a set of inequality constraints, $c_j(x) \geq 0$, $j = 1, \ldots, n_j$; and upper and lower bounds on the continuous variables x_i. Any of the functions involved in both kinds of constraints can be non-linear. In its algebraic form, the general problem is given by Eq. (1)

$$\begin{cases} \min_x f(x) \\ s.t. \quad c_i(x) = 0, \quad i = 1, \ldots, n_i \\ \qquad c_j(x) \geq 0, \quad j = 1, \ldots, n_j \\ \qquad l_i \leq x_i \leq u_i \end{cases} \tag{1}$$

By introducing the slack variables z_j in the inequality constraints, Eq. (1) turns into Eq. (2).

$$\begin{cases} \min_x f(x) \\ s.t. \quad c_i(x) = 0, \quad i = 1, \ldots, n_i \\ \qquad c_j(x) - z_j = 0, \quad j = 1, \ldots, n_j \\ \qquad l_i \leq x_i \leq u_i \\ \qquad z_j \geq 0 \end{cases} \tag{2}$$

Both the objective function and the equality constraints can be combined into an augmented Lagrangian function (Eq. (3) and Eq. (4)), where $\Lambda \in \mathbb{R}^{n_i+n_j}$ is an estimate of the vector of Lagrange multipliers, $\rho > 0$ is the penalty parameter and $\|\cdot\|$ is the Euclidean norm.

$$\mathcal{L}(x, \lambda, \rho) = f(x) + C(x)^T \Lambda + \left(\frac{\rho}{2}\right) \cdot \left\|C(x)\right\|^2 \tag{3}$$

$$C(x) = \{c_i(x)\} \cup \{c_j(x) - z_j\} \tag{4}$$

Eq. (2) is reformulated to Eq. (5) by means of Eq. (3). The problem stated in Eq. (5) becomes box-constrained because the sole explicit constraints are the variable bounds, while the rest of the constraints are embedded in the Augmented Lagrangian.

$$\begin{cases} \min_x \mathcal{L}(x, \Lambda, \rho) \\ s.t. \quad l_i \leq x_i \leq u_i \\ \qquad z_j \geq 0 \end{cases} \tag{5}$$

This change makes it easier to compute the projections onto the feasible region; thus, the general algorithmic performance is improved. This property represents a great advantage for a low-cost algorithm, like the spectral projected gradient method. By virtue of Eq. (5), we shall henceforth refer to the Lagrangian as the objective function of the optimization problem.