INTERVAL MATHEMATICS AND INTERVAL CONSTRAINT PROGRAMMING
ICCM-2004 WORKSHOPS P. 280–284

An interval algorithm combining symbolic rewriting and componentwise Newton method applied to control a class of queueing systems*

Bartlomiej Jacek Kubica†, Krzysztof Malinowski‡

Introduction

The idea of Componentwise Newton Operator has been presented in [9]. It seems up to now not much attention has been paid to this concept.

In this paper we present properties of the Componentwise Newton Operator and show how they can be used in numerical algorithms. It seems they are especially important when transforming the problem by some symbolic methods, based on the computation of Groebner bases.

The resulting algorithm will be applied to a problem of optimizing control rules for a queueing system.

1. Componentwise Newton method

1.1. The definition. According to [9] we define the interval componentwise Newton operator as follows.

Assume we try to solve the equation system:

$$\begin{cases} g_1(x_1, \dots, x_n) = 0, \\ \quad\quad \dots \\ g_n(x_1, \dots, x_n) = 0, \end{cases}$$

where $x_1 \in X_1, \dots x_n \in X_n$.

We denote real variables by small letters and intervals by cardinal letters. Real–valued functions and its interval envelopes are denoted by the same small letters (this should not lead to any misunderstanding. The symbol $\mathrm{mid}(X)$ denotes the midpoint of the interval X.

The componentwise Newton operator for this equation system with respect to box X, equation i and variable j is defined as:

$$N_{\mathrm{cmp}}(X, i, j) = \mathrm{mid}(X_j) - \frac{g_i(X_1, \dots, X_{j-1}, \mathrm{mid}(X_j), X_{j+1}, \dots, X_n)}{\frac{\partial g_i}{\partial x_j}(X_1, \dots, X_n)}. \quad (1)$$

*This work has been supported by the Foundation for Polish Science through its stipends.

†Warsaw University of Technology, e-mail: bkubica@elka.pw.edu.pl
‡Research and Academic Computer Network (NASK), e-mail: K.Malinowski@ia.pw.edu.pl

1.2. Known properties. In [9] two simple properites of the N_{cmp} operator are proven:

Proposition 1. *All roots of the equation system belonging to the box X (if any) must lie in $X \cap N_{\text{cmp}}(X, i, j)$ for any i, j.*

Proposition 2. *If for an arbitrary i and j we have $X \cap N_{\text{cmp}}(X, i, j) = \emptyset$ then there are no roots in the box X.*

1.3. New properties. One of the most interesting and very useful properties of traditional interval Newton operators (wide literatur available, e. g. [10]) is that they allow to verify not only the existence of a root in a box, but also the uniqueness of it. We omit the details due to the fact that they are well-known.

Can analogous properties be found for the componentwise method? The answer is positive, though they hold for special classes of functions only.

Theorem 1. *Assume that for a given i the function $g_i(\cdots)$ depends only on x_j:*

$$g_i(x_1, \ldots, x_n) = g_i(x_j). \tag{2}$$

Then if the componentwise Newton operator satisfies the condition:

$$N_{\text{cmp}}(X, i, j) \subset X_j \tag{3}$$

then there is a unique value of $x_j \in X_j$ for which $g_i(x_1, \ldots, x_n) = 0$. The method has quadratic rate of convergence.

The proof is obvious because for the considered case the componentwise Newton operator is equivalent to the univariate Newton operator of a one–dimensional function.

The condition (2) is not as restrictive as it may seem; if we seek for zeros of the gradient of a separable function, it will be satisfied for all equations.

Can it be somehow applied to non–separable problems, such as the one in section 3, though? The answer is positive again (at least for polynomial equations), but we have to apply some symbolic transformations first.

2. Groebner bases

2.1. Basics. The formal definition of a Groebner basis may be found e. g. in [5] or many other works and books. We will not describe the sophisticated theory here, we just say informally that the computation of a Groebner basis of a polynomial equation system is analogous to transforming the linear equation system to its triangular form.

The meaning is as follows.

Consider an ordering of the variables, say $x_1 \prec x_2 \prec \cdots \prec x_n$. We can transform the polynomial system to an equivalent one, for which the last equation will depend only on x_1, the previous one – on x_1 and x_2 etc. (this property is called *the elimination property* of Groebner bases with lexical ordering).

Groebner bases have already been used in connection with interval analysis (see [3, 4]) and they proved useful. Interval Newton methods were more efficient on transformed equation systems, due to reduced dependency problem.

For componentwise method the problem is a bit different. We would like to have univariate equations for each of the variables, not only for one of them. We can obtain it by computing Groebner bases for different orderings and taking the univariate equations from each of them.

2.2. Conversions of Groebner bases. Unfortunately, the cost of computing a Groebner Basis may be exponential. However, the conversion of one Groebner Basis to another one (e. g. the basis for another ordering of variables) may be done by quite efficient polynomial algorithms.

There are a few methods for such conversions – the FGLM method [8], the Groebner Walk method [6] or the LLL method [2] to name a few.

They differ in many details. In our application most interesting will be the FGLM method. It is restricted to systems with zero-dimensional ideal, but it will allow us to find the univariate equation (for the "smallest" variable) *without* computing the whole basis.

This gives us a fast way to obtain univariate equations for each variable.

2.3. Application to an interval method. The resulting system of n univariate equations won't be equivalent to the primary problem; it will constitute necesssary conditions only. It seems however that using such equations in addition to the original problem may increase the efficiency of the algorithm, because we can quickly reduce the dimensions of boxes by cheap iterations of univariate Newton method.

3. An example – long-run optimization of the Mendelson's queueing system

The considered example is a combination of problems from [7] and [11]. We try to optimize the long-run behaviour of a priority queueing system. The notion "long-run" means in this context that we can control not only the arrival rates $\lambda_1, \ldots, \lambda_R$, but also the service rate μ.

The problem is as follows:

$$\max_{\lambda_1,\ldots,\lambda_R,\mu} J = \sum_{i=1}^{R} V_i(\lambda_i) - \sum_{i=1}^{R} \lambda_i \cdot G_i(\lambda_1, \ldots, \lambda_R; \mu) - C(\mu), \tag{4}$$

s.t.

$$0 \le \lambda_i \le \Lambda_i \quad \forall i = 1, \ldots, R, \tag{5}$$

$$\sum_{i=1}^{R} \lambda_i < \mu. \tag{6}$$

Meaning of the parameters is as follows:

- $V_i(\cdot)$, $i = 1, \ldots, R$, is the aggregate value received by the clients from class i; authors generally assume the form $V_i(\lambda_i) = a_i \cdot \lambda - \dfrac{a_i}{2\Lambda_i} \cdot \lambda_i^2$,

- $G(\cdots)$ is the delay cost (see below),
- $C(\mu)$ is the capacity cost (authors assume linear cost $C(\mu) = b \cdot \mu$).

The problem is known to be nonconvex and multiextremal [12].

In case when delay cost has linear structure, we can write it as:

$$G_i(\lambda_1, \dots, \lambda_R) = \delta \cdot W_i(\lambda_1, \dots, \lambda_R).$$

Mendelson and Whang claim [11] that the expected sojourn times for a non–preemptive priority queue are given by the following equation:

$$W_i(\lambda_1, \dots, \lambda_R) = \frac{\sum_{j=1}^{R} \lambda_j / \mu_j^2}{(1 - \sum_{j=1}^{i-1} \rho_j) \cdot (1 - \sum_{j=1}^{i} \rho_j)} + \frac{1}{\mu_i}, \tag{7}$$

where $\rho_j = \dfrac{\lambda_j}{\mu_j}$.

The motivation for the above formula may be found e. g. in [1].

For homogeneus service times for all client classes, i. e. for $\mu_1 = \ldots = \mu_R = \mu$ equation (7) takes the form:

$$W_i(\lambda_1, \dots, \lambda_R) = \frac{\sum_{j=1}^{R} \lambda_j}{(\mu - \sum_{j=1}^{i-1} \lambda_j) \cdot (\mu - \sum_{j=1}^{i} \lambda_j)} + \frac{1}{\mu}. \tag{8}$$

In the optimization problem formulated above a strict inequality $\sum_{i=1}^{R} \lambda_i < \mu$ is present. Changing it to a non–strict inequality we risk obtaining infinite values of the function during the computations.

We can overcome this difficulty e. g. by the following substitution of the variables:

$$\mu = \sum_{i=1}^{R} \lambda_i + \psi \quad \psi \in [\epsilon_\psi, +\infty]$$

where ϵ_ψ is a small positive number.

In practice the upper bound on the ψ guaranteeing that the maximal value of μ is a few times bigger than the sum of maximal values of the arrival rates is enough in both cases.

The following bounds were used in numerical computations: $\psi \in [0.0005, 5.0]$.

4. Comments

The problem from the previous section is obviously non–separable. All functions may be described by polynomials, though. The developed method seems to be suitable then.

The implementation will be done in C++ and will base on C–XSC 2.0 beta3 and CToolbox 2.0 beta3 (see [14]) and the library from [13] extended by the first author. The symbolic computations use the PoSSoLib package ver. 4.99 [15].

In many cases it may be especially profitable to use the PoSSoLib polynomial coefficients in computations. The coefficients are then polynomials of "variables"

a_i, Λ_i, δ_i. Symbolic computations on them may reduce the overestimation due to occurence of these parameters (usually known only aproximately) in many equations.

Other details of the implementation and numerical results are planned to be presented during the Workshop.

References

[1] Adan I., Resing J. Queueing Theory, 2001, the book downloadable from. – http://www.cs.duke.edu/~fishhai/misc/queue.pdf.

[2] Basiri A., Faugere J.C. Changing the Ordering of Groebner Bases with LLL: Case of Two Variables. – http://www.inria.fr/INRIA/publication/publi-ps-gz/RR/RR-4746.ps.gz

[3] Benhamou F., Granvilliers L. Combining Local Consistency, Symbolic Rewriting and Interval Methods. – 1994. – http://www.sciences.univ-nantes.fr/info/perso/permanents/benhamou/papers/BenGra_AISMC96.pdf.

[4] Benhamou F., Granvilliers L. Automatic Generation of Numerical Redundancies for Non-linear Constraint Solving. – 1997. – http://www.sciences.univ-nantes.fr/info/perso/permanents/benhamou/papers/BenGra_Reliable97.pdf.

[5] Buchberger B. Groebner Bases and Systems Theory. – 2001. – http://www.risc.uni-linz.ac.at/people/buchberg/papers/2001-05-12-A.ps.

[6] Collart S., Kalkenbrener M., Mall D. Converting bases with the Groebner walk // J. Symbolic Computations. – 1997. – No. 24. – P. 465–469.

[7] Dewan S., Mendelson H. User Delay Costs and Internal Pricing for a Service Facility // Management Science. – 1990. – Vol. 36, No. 12. – P. 1502–1517.

[8] Faugere J.C., Gianni P., Lazard D., Mora T. Efficent Computation of Zero-dimensional Groebner Bases by Change of Ordering // J. Symbolic Computations. – 1993. – No. 16. – P. 329–344.

[9] Herbort S., Ratz D. Improving the Efficiency of a Nonlinear-System-Solver Using the Componentwise Newton Method. – 1997. – http://www.uni-karlsruhe.de/~iam/html/reports/rep9702.ps.gz.

[10] Kearfott R.B. An Interval Branch and Bound Algorithm for Bound Constrained Optimization // J. Global Optimization. – 1992. – No. 2. – P. 259–280. – http://interval.louisiana.edu/preprints.html/.

[11] Mendelson H., Whang S. Optimal Incentive – Compatible Priority Pricing for the $M/M/1$ Queue // Operations Research. – 1990. – Vol. 38, No. 5. – P. 870–883.

[12] Stidham Jr. S. Pricing and Capacity Decisions for a Service Facility: Stability and Multiple Local Optima // Management Science. – 1992. – Vol. 38, No. 8. – P. 1121–1139.

[13] Wiechowski P. An object-oriented interval library for solving global optimization problems and nonlinear equation systems (in Polish): BEng Thesis under supervision of Andrzej Karbowski. – Warsaw, 2003.

[14] http://www.xsc.de/.

[15] http://posso.dm.unipi.it/.