2007 International Symposium on Nonlinear Theory and its Applications
NOLTA'07, Vancouver, Canada, September 16-19, 2007

LP Narrowing: A New Strategy for Finding All Solutions of Nonlinear Equations

Koki Suda and Kiyotaka Yamamura

Faculty of Science and Engineering, Chuo University
Tokyo, 112-8551 JAPAN
Email: {suda, yamamura}@yamamura.elect.chuo-u.ac.jp

Abstract—As solution techniques to solve numerical constraint satisfaction problems, consistency techniques such as box consistency are well-known. In this paper, the idea of the box consistency is applied to the problem of finding all solutions of nonlinear equations. A new algorithm is proposed, where boxes that appear in the algorithmic process are narrowed using linear programming such that no solution is lost. It is shown that all solutions can be found very efficiently by the proposed algorithm.

1. Introduction

Many application problems ranging from robotics to chemistry and geometry can be seen as numerical constraint satisfaction problems (NCSPs) [1]–[7]. An NCSP is defined by a set of variables and a set of nonlinear constraints on the variables. The domain of the variables are closed intervals of real values. NCSPs can be used to express a large class of problems including the problem of finding all solutions of nonlinear equations. The goal is to find sharp boxes that approximate the solutions.

Correct approximations of the solutions can be obtained by interval-based solvers; most of them are based on a branch and prune algorithm. This algorithm interleaves domain pruning and domain splitting, until a given precision of the domains is reached. In most interval solvers, the pruning step is based on local consistencies such as box consistency [1]–[7].

In this paper, we introduce the idea of the box consistency to the problem of finding all solutions of a system of nonlinear equations:

$$f(x) = 0 \tag{1}$$

contained in an initial box[1] D in $\mathbb{R}^n$, where $x = (x_1, x_2, \cdots, x_n)^T \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}^n$. The basic idea of the box consistency is narrowing a box (variable domain) as much as possible using interval extension such that no solution is lost. However, the conventional box consistency does not achieve tight narrowing because interval extension often causes overestimating.

In this paper, we combine the idea of the box consistency with the linear programming (LP) techniques proposed in [8]–[10], and propose a new narrowing algorithm termed *LP narrowing*. By introducing the LP narrowing to the interval algorithms such as the Krawczyk-Moore algorithm, all solutions can be found very efficiently.

2. Background of the Study

Many real-world problems require solving NCSPs. An NCSP is a triplet $(\mathcal{V}, \mathcal{C}, \mathcal{D})$ that consists of a finite set $\mathcal{V}$ of variables taking their values in domains $\mathcal{D}$ over the reals and subject to a finite set $\mathcal{C}$ of numerical constraints. A tuple of values assigned to the variables such that all the constraints are satisfied is called a solution. In practice, numerical constraints are often equalities or inequalities expressed in factorable form, that is, they can be represented by elementary functions such as $+, -, \times, \div$, log, exp, sin, $\cos, \cdots$. The problem of finding all solutions of nonlinear equations, which we focus in this paper, is a class of the NCSPs.

Many solution techniques have been proposed to solve NCSPs [1]–[7], and most of them are based on interval arithmetic or its variants [11],[12]. In the last ten years, there have been elaborate uses of interval arithmetic to devise narrowing algorithms, a possible variant is the box consistency. The narrowing algorithm associated with the box consistency returns a sub-box in the initial box that is box-consistent [1]–[7] (or, more informally, the largest sub-box in the initial box that cannot be narrowed further). Actually, since the largest sub-box cannot be obtained in general, its approximation is found by deleting sub-boxes containing no solution that are located in a lower bound and an upper bound of each interval of the initial box by using interval extension[2]. An example of the box consistency when $n = 2$ is illustrated in Fig. 1(a).

However, the conventional box consistency does not

This work was supported in part by the 21st Century COE Program and the Grant-in-Aid for Scientific Research No.17560350 from the Japanese Ministry of Education, Culture, Sports, Science and Technology.

[1] An n-dimensional rectangular region with the sides parallel to the coordinate axes will be called a box.

[2] The interval extension is calculated by replacing the variable x_i with the interval $[a_i, b_i]$ and by replacing the arithmetic operations with the corresponding interval operations [11]. The interval extension is often used as a nonexistence test for solutions because it contains the range of the original nonlinear function.

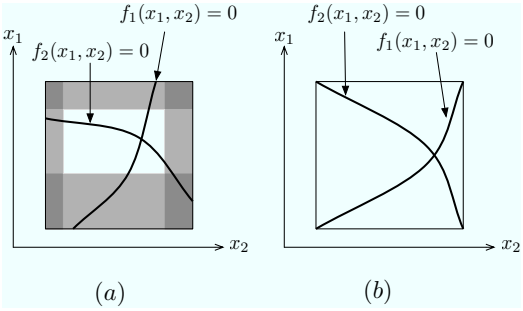

Figure 1: Narrowing a box by box consistecy.

achieve tight narrowing. One of the reasons is that interval extension often causes overestimating. Another reason is that the box consistency handles the constraints independently, so it cannot narrow a box as shown in Fig. 1(b). In this paper, we introduce LP to narrow a box because it handles all the constraints at the same time.

3. Basic Algorithm

In this section, we first summarize the basic procedures of interval algorithms [11],[12].

An n-dimensional interval vector with components $[a_i, b_i]$ $(i = 1, 2, \cdots, n)$ is denoted by

$$X = ([a_1, b_1], [a_2, b_2], \cdots, [a_n, b_n])^T. \tag{2}$$

Geometrically, X is an n-dimensional box.

In interval algorithms, the following procedure is performed recursively, beginning with the initial box $X = D$ [11]. At each level, we analyze the box X. If there is no solution of (1) in X, then we exclude it from further consideration. If there is a unique solution of (1) in X, then we compute it by some iterative method. In the field of interval analysis, computationally verifiable sufficient conditions for nonexistence, existence and uniqueness of a solution in X have been developed. If these conditions are not satisfied and neither existence nor nonexistence of a solution in X can be proved, then split X in some appropriately chosen coordinate direction to form two new boxes; we then continue the above procedure with one of these boxes, and put the other one on a stack for later consideration. Thus, provided the number of solutions of (1) contained in $D \subset \mathbb{R}^n$ is finite, we can find them all with mathematical certainty. This algorithm is a kind of the branch and prune algorithm.

Next, we summarize the powerful nonexistence test proposed in [8]–[10].

For the simplicity of discussion, in this paper we assume that (1) can be represented as

$$f(x) \triangleq Pg(x) + Qx - r = 0 \tag{3}$$

as assumed in [8]–[10], where $g(x) = [g_1(x_1), g_2(x_2), \cdots, g_n(x_n)]^T : \mathbb{R}^n \to \mathbb{R}^n$ is a nonlinear function with component functions $g_i(x_i) : \mathbb{R}^1 \to \mathbb{R}^1$ $(i = 1, 2, \cdots, n)$, P and Q are $n \times n$ constant matrices, and $r = (r_1, r_2, \cdots, r_n)^T \in \mathbb{R}^n$ is a constant vector. Namely, we divide the system into the nonlinear term $Pg(x)$, the linear term Qx, and the constant term r. Note that the discussion in this paper is easily extended to more general systems of nonlinear equations; for details, see [8].

Let the interval extension of $g_i(x_i)$ over $[a_i, b_i]$ be $[c_i, d_i]$. Then, we introduce auxiliary variables y_i and put $y_i = g_i(x_i)$. If $a_i \le x_i \le b_i$, then $c_i \le y_i \le d_i$.

Now we consider the LP problem:

$$\begin{aligned} &\max \ \text{(arbitrary constant)} \\ &\text{subject to} \\ &\quad Py + Qx - r = 0 \\ &\quad a \le x \le b \\ &\quad c \le y \le d \end{aligned} \tag{4}$$

where $y = (y_1, y_2, \cdots, y_n)^T \in \mathbb{R}^n$ and $a \le x \le b$ implies that $a_i \le x_i \le b_i$ for $i = 1, 2, \cdots, n$. Then, we apply the simplex method (two-phase simplex method) to (4).

Evidently, all solutions of (3) that exist in X satisfy the constraints in (4) if we put $y_i = g_i(x_i)$. Hence, if the feasible region of the LP problem (4) is empty, then we can conclude that there is no solution of (3) in X.

The emptiness or nonemptiness of the feasible region of (4) can be checked by the simplex method. If the simplex method terminates with the information that the feasible region is empty, then there is no solution of (3) in X, and we can exclude X from further consideration. This test is called the LP test. The LP test is much more powerful than the conventional test using interval extension because it handles the constraints at the same time. It has been shown that if we use appropriate directed roundings, then the LP test gives correct results (in the sense that boxes containing solutions are never discarded) [8].

By introducing the LP test to the interval algorithms (such as the Krawczyk-Moore algorithm [11]), all solutions of (3) can be found very efficiently. In [8], this algorithm solves a system of nonlinear equations with $n = 60$ in practical computation time, although the original Krawczyk-Moore algorithm can solve the system only for $n \le 12$.

In [9], it is shown that the LP test can be performed with a few iterations (often no iteration) per box by using the dual simplex method. Using this technique, the LP test becomes not only powerful but also efficient. In [9], this improved LP test is introduced to the Krawczyk-Moore algorithm, which succeeded in finding all solutions to systems of nonlinear equations with $n = 200$.

4. Proposed Algorithm

The proposed algorithm is an extension of the algorithm in [9], to which the idea of narrowing a box using LP is introduced. Namely, if X is not excluded, then we narrow the box using LP, which makes the algorithm much more efficient.

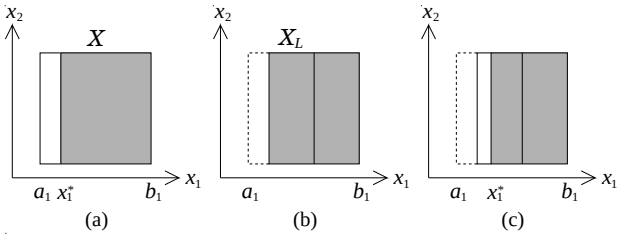

Figure 2: LP narrowing.

We first explain how a box X is narrowed in the x_1-direction. In the LP test to X, we apply the simplex method to

$$\begin{aligned} &\min\ x_i \\ &\text{subject to} \\ &\quad Py + Qx - r = 0 \\ &\quad a \le x \le b \\ &\quad c \le y \le d \end{aligned} \tag{5}$$

for $i = 1$ instead of (4). If the feasible region of (5) is empty, then we exclude X from further consideration. If the minimum value x_1^* is greater than a_1, then we prune the lower part $x_1 \le x_1^*$ of X because there is no solution there [see Fig. 2(a)]. Thus, the box X is narrowed from the lower side of the x_1-direction.

Now we split the narrowed box in the x_1-direction [see Fig. 2(b)]. Let the lower sub-box be denoted by X_L. Then, the following procedure is repeated beginning with this sub-box. We solve the LP problem (5) for X_L (where $a \le x \le b$ and $c \le y \le d$ represent the corresponding intervals for X_L). Note that this LP problem can be solved efficiently with a few iterations by using the dual simplex method. If the feasible region of (5) is empty, then we exclude X_L from further consideration. If the minimum value x_1^* is greater than the lower bound of X_L in the x_1-direction, then we prune the lower part $x_1 \le x_1^*$ of X_L [see Fig. 2(c)], split the narrowed box in the x_1-direction, and let the lower sub-box be denoted by X_L again. This procedure is repeated until there is no feasible region of (5) and hence X_L is excluded. Thus, the box X is narrowed repeatedly from the lower side of the x_1-direction.

Let the resulting narrowed box be denoted by X again for the simplicity of notation. Then, we apply the simplex method to

$$\begin{aligned} &\max\ x_i \\ &\text{subject to} \\ &\quad Py + Qx - r = 0 \\ &\quad a \le x \le b \\ &\quad c \le y \le d \end{aligned} \tag{6}$$

for $i = 1$ on X. If the maximum value x_1^* is less than b_1, then we prune the upper part $x_1 \ge x_1^*$ of X, split the narrowed box in the x_1-direction, and let the upper sub-box be denoted by X_U. Then, the similar procedure is repeated until there is no feasible region of (6) and hence X_U is excluded. Thus, the box X is narrowed repeatedly from the upper side of the x_1-direction.

This is the narrowing procedure in the x_1-direction. Let the resulting narrowed box be denoted by X again. Then, we repeat the similar narrowing procedure in the x_i-directions ($i = 2, 3, \cdots, n$), and narrow the box in all coordinate directions. Such an algorithm is called the LP narrowing.

Note that as the box becomes smaller, the feasible region becomes smaller, which makes the LP test more powerful. Also note that the LP problem (5) or (6) can be solved efficiently with a few iterations by the dual simplex method. Thus, the LP narrowing is not only powerful but also efficient and narrows a box as much as possible using the dual simplex method.

Table 1: Comparison of computation time (s) in Example 1.

n	Ref.[9]	Proposed
50	1 808	7
100	5 914	47
150	8 777	151
200	34 702	361
250	17 8984	508
300	–	1 194
350	–	1 729
400	–	1 957
450	–	3 944
500	–	3 817
550	–	6 588
600	–	9 530
650	–	10 948
700	–	15 021

5. Numerical Examples

We introduced the LP narrowing to the well-known Krawczyk-Moore algorithm [11] and implemented the new algorithms using the programming language C on a Sun Ultra 45 workstation (CPU: UltraSPARC-IIIi 1.6GHz). In this section, we show some numerical examples.

We used the free package GLPK (GNU Linear Programming Kit)[3] for solving the LP problem (4), (5), or (6). The GLPK package is a set of routines written in ANSI C and organized in the form of a callable library. This package is intended for solving large-scale LP, mixed integer LP, and other related problems. The main advantage of using GLPK in the proposed algorithm is that GLPK can perform the dual simplex method starting from a previously obtained dual feasible basis. Hence, GLPK is well-suited to the proposed algorithm.

[3] http://www.gnu.org/software/glpk/

Table 2: Comparison of computation time (s) in Example 2.

n	Ref.[9]	Proposed
50	117	16
100	1 514	105
150	9 654	426
200	39 883	1 158
250	84 715	2 353
300	–	4 901
350	–	7 036
400	–	11 230
450	–	15 292
500	–	25 125
550	–	30 374
600	–	37 635
650	–	47 567
700	–	63 193
750	–	75 419
800	–	99 392
850	–	110 802
900	–	132 449
950	–	170 198
1000	–	178 707

Example 1: We first consider a system of n nonlinear equations:

$$x_{i+1} - 2x_i + x_{i-1} + h^2 \exp(x_i) = 0, \quad i = 1, 2, \cdots, n$$

where $x_0 = x_{n+1} = 0$ and $h = 1/(n+1)$. This system comes from a nonlinear two-point boundary value problem termed the Bratu problem. The initial box is $D = ([0, 5], \cdots, [0, 5])^T$. The number of solutions is two for all n.

Table 1 compares the computation time (s) of the algorithm in [9] and the proposed algorithms, where "–" implies that it could not be computed in practical computation time. As seen from the table, the proposed algorithm is much more efficient than the algorithm in [9], which indicates that the LP narrowing is very powerful.

Example 2: We next consider a system of n nonlinear equations:

$$x_i - \frac{1}{2n}\left(\sum_{j=1}^{n} x_j^3 + i\right) = 0, \quad i = 1, 2, \cdots, n.$$

The initial box is $D = ([-10, 10], \cdots, [-10, 10])^T$. The number of solutions is three for all n. Table 2 compares the computation time (s) of the algorithm in [9] and the proposed algorithms. It is seen that a similar result is obtained as that in Example 1.

6. Conclusion

In this paper, an efficient algorithm has been proposed for finding all solutions of nonlinear equations using the concept of LP narrowing. It has been shown that all solutions can be found very efficiently by the proposed algorithm.

As has been stated, the problem of finding all solutions of nonlinear equations is a class of NCSPs. It seems that the LP techniques are also effective for a more general class of NCSPs. Hence, it should be left as a future subject to apply the LP techniques to a more general class of NCSPs.

References

[1] O. Lhomme, "Consistency techniques for numeric CSPs," Proc. Int. Joint Conf. Artificial Intelligence, vol.1, Chambery, France, pp.232–238, Aug. 1993.

[2] F. Benhamou, D. McAllester, and P.V. Hentenryck, "CLP(Intervals) revisited," Proc. Int. Symp. Logic Programming, Ithaca, NY, pp.124–138, Nov. 1994.

[3] P.V. Hentenryck, D. McAllester, and D. Kapur, "Solving polynomial systems using a branch and prune approach," SIAM J. Numerical Analysis, vol.34, no.2, pp.797–827, April 1997.

[4] H. Collavizza, F. Delobel, and M. Rueher, "Comparing partial consistencies," Reliable Computing, vol.5, no.3, pp.213–228, Aug. 1999.

[5] F. Benhamou, F. Goualard, L. Granvilliers, and J.-F. Puget, "Revising hull and box consistency," Proc. Int. Conf. Logic Programming, Las Cruces, New Mexico, pp.230–244, Nov. 1999.

[6] L. Granvilliers, F. Goualard, and F. Benhamou, "Box consistency through weak box consistency," Proc. IEEE Int. Conf. Tools with Artificial Intelligence, Chicago, IL, pp.373–380, Nov. 1999.

[7] X.-H. Vu, "Rigorous solution techniques for numerical constraint satisfaction problems," Ph.D. Thesis no. 3155, Swiss Federal Institute of Technology in Lausanne, (Artificial Intelligence Laboratory), March 2005.

[8] K. Yamamura, H. Kawata, and A. Tokue, "Interval solution of nonlinear equations using linear programming," BIT—Numerical Mathematics, vol.38, no.1, pp.186–199, Jan. 1998.

[9] K. Yamamura and T. Fujioka, "Finding all solutions of systems of nonlinear equations using the dual simplex method," Proc. Int. Symp. Nonlinear Theory and its Applications, Zao, Japan, pp.219–222, Oct. 2001.

[10] K. Yamamura and K. Suda, "An efficient algorithm for finding all solutions of separable systems of nonlinear equations," BIT—Numerical Mathematics, vol.47, no.3, Sept. 2007.

[11] R. E. Moore, Methods and Applications of Interval Analysis, SIAM Studies in Applied Mathematics, Philadelphia, 1979.

[12] L. Jaulin, M. Kieffer, O. Didrit, and É. Walter, Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics, Springer, London, 2001.