Piotr's Writeup

Piotr Wojciechowski
LCSEE,
West Virginia University,
Morgantown, WV
{pwojciec@mix.wvu.edu}

May 29, 2014

1 May 5, 2014

Looked into methods of reducing the bound in Nelson's paper to reduce the upper limit on the number of break points.

2 April 28, 2014

Focused on work for other classes, no progress.

3 April 21, 2014

Focused on work for other classes, no progress.

Grade: I.

4 April 14, 2014

Nelson's paper describes a $O(m \cdot n^{\lceil \log_2 n \rceil + 3} \cdot \log n)$ time algorithm for solving systems of TVPI constraints. While not as efficient as Hochbaum's algorithm this algorithm keeps track of all generated constraints, except those which are redundant. This makes it more useful to our problem since we are generating the complete upper and lower envelopes for each pair of variables. Thus, we know that the bounds obtained by this algorithm can be applied to our problem.

While the general algorithm is not polynomial, any sub-case where there are a fixed number of coefficients can be solved in polynomial time. Since out problem is simpler than TVPI constraints, we are only dealing with monotone constraints, it may be possible to obtain a more efficient bound. If such a bound is found then our method for pre-computing closure can run in polynomial time.

Grade: C

5 April 7, 2014

Fell ill, no progress.

Grade: I

6 March 31, 2014

Read Hochbaum's paper to determine how to prevent the number of breakpoints from growing exponentially. Hochbaum's for eliminating a variable x_i compares the breakpoints derived from existing constraints involving x_i to try and find a breakpoint inside the feasible range of x_i. If there is a breakpoint in that range then x_i is assigned the value it takes at that breakpoint. Otherwise, two adjacent breakpoints are found such that the entire feasible range of x_i lies between their x_i values. In this case, for each x_j there are at most two non-redundant x_i, x_j constraints over x_is feasible range. This is because if there were more then there would be a breakpoint inside that range.

Hochbaum's algorithm correctly simplifies the Fourier Motzkin elimination procedure to polynomial time. However this simplification is of no use to our problem because only one breakpoint is kept even if multiple breakpoints lie in the feasible interval of x_i.

Grade: C

7 March 24, 2014

The method shown in the March, 17 section only runs in polynomial time if the Fourier-Motzkin elimination procedure can be modified to run in polynomial time for generalized difference constraints. Hochbaum does this by only keeping the non-redundant constraints at each elimination step. We can constantly maintain the sequences of breakpoints for each pair of variables x_i, x_j and then use this sequence to determine which constraints are redundant after each elimination step.

In our sequence of breakpoints if a newly derived x_i, x_j constraint creates new breakpoints then it must also eliminate at least one existing breakpoint. If the constraint did not, then every existing breakpoint would satisfy it and so the constraint would be redundant and not need to be added to the system. Thus since adding a constraint results in at most two new breakpoints (one at each end of where it it the bounding constraint in the x_i, x_j plane) the number of breakpoints wil increase by at most one.

If adding the constraint removes more than one existing breakpoint then the breakpoints removed must be adjacent. Thus at least one of the existing constraints in the system will loose both of its associated breakpoints and so become redundant. This means that it can be safely removed from the system without affecting feasibility. In these cases adding the new constraint will not increase, but might decrease, both the number of breakpoints and non-redundant constraints in the system.

Thus from the list of breakpoints for x_i and x_j we can easily determine which constraints are made redundant by the addition of a new x_i, x_j constraint.

The only constraints added to the system which result in an overall increase in both the number of breakpoints and the number of non-redundant constraints are those that result in the removal of exactly one existing breakpoint.

Grade: C

8 March 17, 2014

For a given system of generalized difference constraints, $\mathbf{A} \cdot \mathbf{x} \geq \mathbf{b}$ let us focus on studying the change in $\min a \cdot x_i - b \cdot x_j$, given x_j as a goes from 0 to ∞. We will start with $a \gg b$ and study how the optimum value changes as a decreases. Thus we will study how the optimum changes as we increase the slope of the objective function.

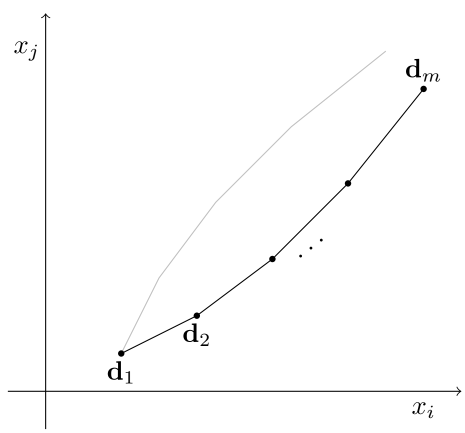
Since our current objective function contains only x_i and x_j we can use Fourier-Motzkin elimination to collapse the original polytope onto the x_i, x_j plane. Let $\mathbf{A}' \cdot \mathbf{x}' \geq \mathbf{b}'$ be the result of this elimination process with all redundant constraints removed.

Each constraint bounding $\mathbf{A}' \cdot \mathbf{x}' \geq \mathbf{b}'$ from below has the following form $a_j \cdot x_i - b_j \cdot x_j \geq c_j$. We can order these constraints by their slope, $\frac{b_j}{a_j}$ in the x_i, x_j plane. Thus we will have that $\frac{b_1}{a_1} < \frac{b_2}{a_2} < \ldots < \frac{b_m}{a_m}$. We are guaranteed strict inequalities because re removed all redundant constraints.

From these constraint we can define the points $\mathbf{d}_1, \ldots, \mathbf{d}_m$ as follows:

1. $\mathbf{d}_1$ is the least element of $\mathbf{A}' \cdot \mathbf{x}' \geq \mathbf{b}'$
2. for each $k = 2, \ldots, m$, $\mathbf{d}_k$ is the intersection of $a_{k-1} \cdot x_j - b_{k-1} \cdot x_j = c_{k-1}$ and $a_k \cdot x_j - b_k \cdot x_j = c_k$

Thus, the lower boundary of the solution space to $\mathbf{A}' \cdot \mathbf{x}' \leq \mathbf{b}'$ will look something like this.



When $a \gg b$, so that $\frac{b}{a} < \frac{b_1}{a_1}$, we have that the optimum solution to

$$\min a \cdot x_i + b \cdot x_j$$
$$\mathbf{A}' \cdot \mathbf{x}' \geq \mathbf{b}'$$

occurs at the least element of that polytope, $\mathbf{d}_1$. As we decrease a, thus increasing the slope of the optimization function we have that as long as $\frac{b}{a} \leq \frac{b_1}{a_1}$ then the optimum solution still occurs at $\mathbf{d}_1$. When $\frac{b}{a} = \frac{b_1}{a_1}$ then the optimum solution occurs at all points between $\mathbf{d}_1$ and $\mathbf{d}_2$. When $\frac{b_1}{a_1} < \frac{b}{a} \leq \frac{b_2}{a_2}$ then the optimum point occurs at $\mathbf{d}_2$. We actually have that while $\frac{b_{k-1}}{a_{k-1}} < \frac{b}{a} \leq \frac{b_k}{a_k}$ then the optimum solution occurs at $\mathbf{d}_k$. If $\frac{b}{a} > \frac{b_m}{a_m}$ and the polytope is bounded then the optimum solution will occur at the greatest? element of $\mathbf{A}' \cdot \mathbf{x}' \leq \mathbf{b}'$ which we will call $\mathbf{d}_{m+1}$. However if the polytope is unbounded then no optimum solution will exist.

Grade: B

9 March 10, 2014

9.1 Relation to generalized flows

We have that the constraint $a_i \cdot x_i - b_j \cdot x_j \leq c$ is equivalent to the constraint $x_i - \frac{b_j}{a_i} \cdot x_j \leq \frac{c}{a_i}$. Thus the entire generalized difference constraint system, $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}$, can be re written as $\mathbf{A}' \cdot \mathbf{x} \leq \mathbf{b}'$ where each constraint has the form $x_i - a \cdot x_j \leq b$, where $a \in \mathbb{Q}^+$ and $b \in \mathbb{Q}$.

Such a system of constraints is similar to a generalized flow network. In a generalized flow network an input of s into a pipe does not guarantee the output to also be s. In such a network each pipe outputs either a fraction or multiple of the amount put in. However, The linear program associated with a generalized flow network is not a system of generalized difference constraints.

9.2 Relation to Meggido's paper

Meggido's paper presents two algorithms for solving systems of TVPI constraints. The first of these algorithms is deterministic and runs in $O(m \cdot n^2 \cdot (\log m + \log^2 n))$ time and can run in parallel on $O(m \cdot n)$ processors with $O(n \cdot (\log m + \log^2 n))$ time on each processor. The second algorithm is randomized and has an expected running time of

$$O(n^3 \cdot \log n + m \cdot n \cdot (\log^5 n + \log m \cdot \log^3 n))$$

This algorithm can be run in parallel on $O(n^2 + m)$ processors with $O(n \cdot (\log^5 n + \log m \cdot \log^3 n))$ time on each. However, neither of these algorithms is for the closure problem.

9.3 Answering Chandra's query

The method proposed by Dr. Chandrasekaran is a way to convert certain forms of optimization problems over a generalized version of horn constraints into finding the least element of a related system of constraints.

Finding

$$\begin{aligned} \min a_i \cdot x_i - b_j \cdot x_j \\ \mathbf{A}' \cdot \mathbf{x} \geq \mathbf{b}' \end{aligned}$$

where $\mathbf{A} \cdot \mathbf{x} \geq \mathbf{b}$ is a system of generalized difference constraints, is a problem of the form required for the method to work. However, the method proposed requires the value x_j^* which is the value assumed by x_j in the optimal solution to

$$\begin{aligned} \min a_i \cdot x_i - b_j \cdot x_j \\ \mathbf{A}' \cdot \mathbf{x} \geq \mathbf{b}' \end{aligned}$$

Grade: C

10 March 3, 2014

10.1 Specification of closure problem

Definition 10.1 *A system of* generalized difference constraints *is a system of constraints* $\mathbf{A} \cdot \mathbf{x} \geq \mathbf{b}$ *where each constraint is of the form* $a_i \cdot x_i - b_j \cdot x_j \geq c$ *where* $a_i, b_j \in \mathbb{Z}^+$ *and* $c \in \mathbb{Z}$.

We can also specify the optimization version of this problem as follows.

Definition 10.2 *In the optimization version of a system of generalized difference constraints we are given the system* $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}$ *of constraints and asked to find*

$$\begin{array}{c} \min a_i \cdot x_i - b_j \cdot x_j \\ \mathbf{A}' \cdot \mathbf{x} \geq \mathbf{b}' \end{array}$$

for some $a_i, b_j \in \mathbb{Z}^+$.

From this we can then specify the definition of the closure of a system of generalized difference constraints.

Definition 10.3 *A system* $\mathbf{A}' \cdot \mathbf{x} \leq \mathbf{b}'$ *is the closure of the generalized difference constraint system* $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}$ *if*

a) $\forall a_i, b_j \in \mathbb{Z}^+$ *we have that* $\begin{array}{c} \min a_i \cdot x_i - b_j \cdot x_j \\ \mathbf{A}' \cdot \mathbf{x} \leq \mathbf{b}' \end{array} = \begin{array}{c} \min a_i \cdot x_i - b_j \cdot x_j \\ \mathbf{A} \cdot \mathbf{x} \leq \mathbf{b} \end{array}$.

b) Finding $\begin{array}{c} \min a_i \cdot x_i - b_j \cdot x_j \\ \mathbf{A}' \cdot \mathbf{x} \leq \mathbf{b}' \end{array}$ *for each* a_i *and* b_j *can be done quickly.*

10.2 Relation to Hochbaum's work

Hochbaum's work deals with TVPI systems. That is systems of constraints in which each constraint involves two variables. The problem of generalized difference constraints is a sub-problem of this. Namely a system of generalized difference constraints is exactly a monotone system of TVPI constraints. A monotone system is one in which each constraint has both a positive and a negative coefficient. For this problem the algorithm in the paper obtains a running time of $O(m \cdot n^2 \cdot \log m)$.

While Hochbaum does not study the optimization or closure problems for the linear case, the paper does mention that the running time of such optimization algorithms is strongly polynomial when the number of variables in the objective function is fixed. Since we are only concerned with optimization functions which have two variables we already know that a single bound can be found in polynomial time.

Thus our work differs from that done by Hochbaum.

10.3 Optimization

Consider the system $\mathbf{A} \cdot \mathbf{x} + c \cdot x_j \cdot \mathbf{a_i} \leq \mathbf{b}$. Let $\mathbf{x}' = (x'_1, \ldots, x'_i, \ldots, x'_n)$ be a solution to $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}$. We can construct the vector $\mathbf{x}'' = (x'_1, \ldots, x'_i - c \cdot x'_j, \ldots x'_n)$. Thus for each $k \neq i$, we have that $x''_k = x'_k$ and that $x''_i = x'_i - c \cdot x'_j$. When we plug $\mathbf{x}''$ into the system $\mathbf{A} \cdot \mathbf{x} + c \cdot x_j \cdot \mathbf{a_i} \leq \mathbf{b}$ we get that

$$\mathbf{A} \cdot \mathbf{x}'' + c \cdot x''_j \cdot \mathbf{a_i} = \mathbf{A} \cdot \mathbf{x}' - c \cdot x'_j \cdot \mathbf{a_i} + c \cdot x'_j \cdot \mathbf{a_i} = \mathbf{A} \cdot \mathbf{x}' \leq \mathbf{b}$$

Thus if $\mathbf{x}'$ is a solution to $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}$ then $\mathbf{x}''$ is a solution to $\mathbf{A} \cdot \mathbf{x} + c \cdot x_j \cdot \mathbf{a_i} \leq \mathbf{b}$. Thus we have that finding

$$\begin{array}{c} \min x_i \\ \mathbf{A} \cdot \mathbf{x} + c \cdot x_j \cdot \mathbf{a_i} \leq \mathbf{b} \end{array}$$

is equivalent to finding

$$\begin{array}{c} \min x_i - c \cdot x_j \\ \mathbf{A} \cdot \mathbf{x} \leq \mathbf{b} \end{array}$$

However, this new system is not necessarily a generalized difference constraint system, or even a TVPI system.

Grade: B.

11 Feb. 24, 2014

We are interested in the closure problem for systems of generalized difference constraints. A *difference constraint* is a constraint of the form $x_i - x_j \leq c_{ij}$. A generalized difference constraint is formed when we allow non-unit coefficients on the variables thus obtaining a constraint of the form $a \cdot x_i - b \cdot x_j \leq c_{ij}$ where $a, b \in \mathbb{Z}^+$. To solve the closure problem for such a system we wish to develop an algorithm that pre-processes a system of generalized difference constraints so that for any $a, b \in \mathbb{Z}^+$ and i and j we can efficiently find $\min a \cdot x_i - b \cdot x_j$ and $\max a \cdot x_i - b \cdot x_j$.

A system of generalized difference constraints is simply a system of monotone TVPI constraints. Thus the problem we are studying is related to the optimization problem over monotone systems of TVPI constraints. We cannot directly calculate $\min$ and $\max$ for all constraints of the form $a \cdot x_i - b \cdot x_j$, thus we will need to utilize the methods used to solve the optimization version of monotone TVPI constraints to see how we can modify the original system to accelerate the process of finding the optimum value of a monotone TVPI objective function.

Grade: C.

12 Feb. 17, 2014

Fell ill, no progress.

Grade: I.

13 Feb. 10, 2014

The Hochbaum paper covers various problems related to systems of two variable per inequality (TVPI) constraints. The paper covers both the linear and integer cases. In the linear case a modified version of Fourier-Motzkin elimination is used to provide a $O(m \cdot n^2 \cdot \log m)$ algorithm for determining the linear feasibility of a system of TVPI constraints. The paper does not provide an optimization algorithm for this problem. However, the paper does mention that the best known optimization algorithms are exponential in the number of variables in the optimization function. For the integer case the Hochbaum paper provides a pseudo-polynomial algorithm for the optimization problem when the system is monotone.

Grade: D.