Projective methods for constraint satisfaction and global optimization

Arnold Neumaier, Ferenc Domes, Hermann Schichl, Mihaly Markot
(University of Vienna, Austria)
Many constraint satisfaction problems

\[
\text{find } x \in x \text{ with } F(x) \in F
\]

and global optimization problems

\[
\min f(x) \text{ s.t. } x \in x, \ F(x) \in F
\]

contain some unbounded variables.

Their solution by branch and bound methods poses special challenges as the search region is infinitely extended.
Most branch and bound solvers add artificial bounds to make the problem bounded, or require the user to add these.

However, if these bounds are too small, they may exclude a solution, while when they are too large, the search in the resulting huge but bounded region may be very inefficient.

Moreover, if the global solver must provide a rigorous guarantee (as for the use in computer-assisted proofs), such artificial bounds are not permitted without justification.
We developed methods based on compactification and projective geometry to cope with the unboundedness in a rigorous manner.

Two different versions of the basic idea, namely

(i) projective constraint propagation, and
(ii) projective transformation of the variables,

are implemented in the rigorous global solvers COCONUT and GloptLab.
Numerical tests demonstrate the capability of the new technique, combined with standard pruning methods, to rigorously solve unbounded global problems.
Projective constraint propagation

We represent each variable and intermediate expression $x_k$ in the form

$$x_k = \hat{x}_k / t^{m_k},$$

where $m_k$ is a rational number and $t$ a scaling factor to be chosen.

The new variable $t$ and the exponents $m_k$ are defined such that $t \in [0, 1]$ and all $\hat{x}_k$ are well-bounded.

This is achieved by a recursive construction, implemented in a walk through a computational DAG.
For the original variables $x_k$ and all interval constants, we define

$$m_k = 0 \text{ if } x_k \text{ is well-bounded},$$
$$m_k = 1 \text{ if } x_k \text{ is unbounded or has a huge bound}.$$

$K$ denotes the set of indices of original variables with $m_k = 1$. The choice

$$t := \left(1 - s + \sum_{k \in K} d_k x_k^2\right)^{-1/2}$$

with $s \leq 1$ and scaling factors $d_k > 0$ leads to the constraint

$$(1 - s)t^2 + \sum d_k \hat{x}_k^2 = 1,$$

from which we deduce the bounds

$$t \in \left[0, \max(0, 1 - s)^{-1/2}\right],$$
$$|\hat{x}_k| \leq d_k^{-1/2} \text{ for } k \in K.$$
To guarantee that \( t \) is real, we assume that
\[
\sum_{k \in K} d_k x_k^2 \geq s
\]
is a valid constraint. For example,
\[
s := \inf \sum_{k \in K} d_k x_k^2
\]
qualifies (if necessary, rescale the \( d_k \) to have \( s \leq 1 \)), but better bounds might be available. Then
\[
t \in [0, 1].
\]

Since \( \hat{x}_k = x_k \) for the well-bounded variables, we have expressed all variables in terms of bounded ones.
The exponents $m$ for an intermediate variable $x$ depend on the operation that creates it.

If $x = \sum \alpha_j x_j$ then $x = \hat{x}/t^m$ with

$$m := \max m_j, \quad \hat{x} := \sum \alpha_j t^{m-m_j} \hat{x}_j,$$

and we get the finite enclosure

$$\hat{x} \in \hat{x} := \sum \alpha_j [0, 1]^{m-m_j} \hat{x}_j.$$

If $x = \prod x_j^{\alpha_j}$ with rational $\alpha_j$ then $x = \hat{x}/t^m$ with

$$m := \sum \alpha_j m_j, \quad \hat{x} := \prod \hat{x}_j^{\alpha_j},$$

and we get the finite enclosure

$$\hat{x} \in \hat{x} := \prod \hat{x}_j^{\alpha_j}.$$

This accounts for all elementary operations and powers with fixed exponent.
For other elementary functions, one can derive similar formulas, though their derivation and implementation is more complex.

For example, if \( x = \log x_j \) then \( x = \hat{x}/t^m \) with

\[
m := 1, \quad \hat{x} := t(\log \hat{x}_j - m \log t),
\]

and we get a finite enclosure derivable by monotony considerations.

These operations were implemented in the COCONUT global optimization platform.
Proceeding with all intermediate expressions, one transforms each constraint of the form \( l(x) = r(x) \) or the form \( l(x) \in r \) into an equation of the form \( x_i = x_j \).

Constraint propagation on the equivalent equation

\[
\hat{x}_i = \hat{x}_j t^{m_i - m_j}
\]

may reduce the bounds for the variables \( x_i, x_j, \) and \( t \).

The improvements can be propagated in the same manner to all intermediate results in a backward sweep and ultimately to the original variables.

The COCONUT constraint propagation on DAGs is even more sophisticated and avoids unnecessary evaluation steps in repeated sweeps.
From the COCONUT test set, 849 global optimization and constraint satisfaction problems of dimensions \( \leq 50 \) were considered. 675 of these contained at least one unbounded variable.

We used local optimization to find a local minimum of the objective (if one was present), giving an additional constraint \( f(x) \leq f_{\text{best}} \).

To test the efficiency of constraint propagation, we turned off all other range reduction methods.

To handle the unbounded part of the search region, we added the further constraint \( \sum x_i^2 \geq 1000 \), defining the **outer problem**. The complementary constraint defines a well-bounded **inner problem** (usually containing all global solutions), and needs no projective treatment.
171 of the 675 unbounded outer problems may be considered as trivial, as they were proved infeasible by standard constraint propagation.

This leaves 504 nontrivial outer problems. Many of them are expected to be infeasible, so we simply tested infeasibility.

Using projective constraint popagation alone proved 85 nontrivial outer problems (17%) infeasible (and 168 trivial ones = 98%).

Combining standard CP with projective CP, performing an internal intersection after each operation, proved 269 of the 504 nontrivial outer problems (53%) infeasible.

Thus projective constraint propagation is a valuable enhancement of the CP toolkit.
Projective transformation to bounded form

GLOPTLab is a constraint satisfaction package for enclosing all solutions of systems of quadratic equations and inequalities.

The special quadratic structure allows one to implement projective transformations explicitly by rewriting the original equations after a projective transformation

\[ x_i = \frac{\hat{x}_i}{t} \]

of all variables \( x_i \), where (with positive constants \( c \) and \( d_i \))

\[ t := \left( c + \sum_i d_i x_i^2 \right)^{-1/2}. \]
Linear inequality constraints

\[ Ax \geq b \]

are transformed into the homogeneous linear constraints

\[ A\hat{x} - bt \geq 0. \]

Equations and inequalities with the opposite sign are handled in the same way.

Two-sided linear constraints are either treated as two separate linear inequalities, or as equations with a bounded slack variable.

Bound constraints are treated as linear constraints, too.
Nonlinear quadratic constraints

\[ x^T G x + c^T x \geq \gamma \]

are transformed into the homogeneous quadratic constraints

\[ \hat{x}^T G \hat{x} + t c^T \hat{x} - \gamma t^2 \geq 0. \]

Again, equations and inequalities with the opposite sign are handled in the same way. Two-sided quadratic constraints are treated as equations with a bounded slack variable.

An analogous transformation works for arbitrary polynomial constraints, and even for signomials (linear combinations of power products with rational exponents).
The definition of the transform gives the constraint

\[ ct^2 + \sum_i d_i \hat{x}_i^2 = 1, \]

which immediately implies the bound constraints

\[ t \in [0, c^{-1/2}], \quad \hat{x}_i \in [-d_i^{-1/2}, d_i^{-1/2}]. \]

Thus the transformed problem is again quadratic and bounded. Hence there is no problem solving it with the traditional methods.

An advantage of this transformation is that all range reduction methods can be applied to this formulation, not only CP.
After having solved the transformed problem, one can recover the original solution from

\[ x_i = \frac{\hat{x}_i}{t}. \]

Alternatively, one can solve a bigger constraint satisfaction problem containing both the original and the transformed variables and constraints.

In this case, one must add the transformation itself as additional quadratic constraints

\[ x_i t - \hat{x}_i = 0. \]

This allows one to exploit both the features of the original and of the transformed problem at the same time, at the cost of doubling the problem size.
If an objective function is present, its nonlinear part must be substituted by a new variable and a quadratic equality constraint.

Then the problem has a linear objective function $c^T x$.

The projectively transformed problem then has as objective a single, additional variable $s$ and the additional quadratic constraint

$$st - c^T \hat{x} = 0.$$
Example. The constraint satisfaction problem

\[0.36x_1 - x_2 = 0.75,\]
\[2x_1^2 - x_2^2 = 1,\]
\[x_1 \geq 0, \quad x_2 \geq 0\]
cannot be solved with standard constraint propagation and branch and bound, since no box \(x\) of the form \(x_1 = [a, \infty], x_2 = [b, \infty]\) can be eliminated by CP.

The problem is infeasible but the equations have a solution at

\[x_1 \approx 0.6491, \quad x_2 \approx -0.5163,\]

slightly outside the defining box.
Using \( t = (1 + x_1^2 + x_2^2)^{-1/2} \), the projective transform of this problem becomes

\[
.36\hat{x}_1 - \hat{x}_2 = .75t, \tag{1}
\]
\[
2\hat{x}_1^2 - \hat{x}_2^2 = t^2, \tag{2}
\]
\[
\hat{x}_1 \geq 0, \quad \hat{x}_2 \geq 0, \tag{3}
\]
\[
t^2 + \hat{x}_1^2 + \hat{x}_2^2 = 1. \tag{4}
\]

From (3) and (4), we have \( \hat{x}_1, \hat{x}_2, t \in [0, 1] \).

(1) then implies \( \hat{x}_2 \leq .36 \) and \( t \leq .48 \).

Now (4) gives \( \hat{x}_1^2 \geq 1 - .36^2 - .48^2 = .64 \), hence \( \hat{x}_1 \geq .8 \).

Finally, the left hand side of (2) is

\[
2\hat{x}_1^2 - \hat{x}_2^2 \geq 2 \times .8^2 - .36^2 = 1.1504, \]

while the right hand side is \( t^2 \leq .48^2 = .2304 \).

This implies that (2) cannot be satisfied.
Thus CP proved infeasibility of the transformed problem, and hence of the original problem.
Slides of this talk
www.mat.univie.ac.at/~neum/projslides.pdf

Global (and Local) Optimization
www.mat.univie.ac.at/~neum/glopt.html

The COCONUT environment
http://www.mat.univie.ac.at/~coconut/coconut-environment/

GloptLab
http://www.mat.univie.ac.at/~dferi/gloptlab.html