Solving the Multiple Constant Multiplication Problem with Constraint Programming

Théo Cantaloube
INSA Lyon, CITI, UR3720, 69621 Villeurbanne, France

Xiao Peng
LAAS-CNRS, University of Toulouse, Toulouse, France

Christine Solnon
INSA Lyon, Inria, CITI, UR3720, 69621 Villeurbanne, France

Anastasia Volkova
Inria, INSA Lyon, CITI, UR3720, 69621 Villeurbanne, France

Abstract

The Multiple Constant Multiplication (MCM) problem arises in many applications such as, for example, digital signal processing or deep neural network inference. Given a set T of target constants, the goal is to find the most efficient way for multiplying an input number with every constant in T. Multiplications are done by combining additions/subtractions with bit-shifts (shifting X by k bits on the left or the right is equivalent to multiplying X by 2^k or 2^{-k}). Additions/subtractions are called adders, and the goal is to minimize the number of adders because bit-shifts are basically wiring in hardware, and are considered as free in comparison with adders. A key point is that intermediate results (called fundamentals) may be shared to produce different target constants. For example, when $T = \{7, 19, 93\}$, an optimal solution with 4 adders is: $7X = 2^3X - X; 31X = 2^5X - 5X; 19X = 2^{-1}(7X + 31X); 93X = 31X + 2^1 \cdot 31X$. In this solution, the fundamental 31 is used to produce both 19 and 93. The corresponding adder graph is displayed in Fig. 1.

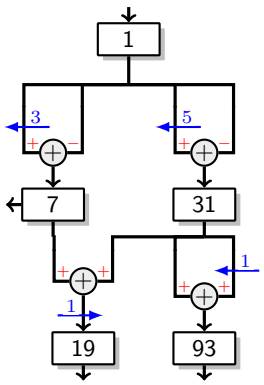

Figure 1 Optimal adder graph for $T = \{7, 19, 93\}$

The MCM problem is generally considered as NP-hard and state-of-the-art approaches use Integer Linear Programming (ILP) [2] or SAT [1] models. The challenge both lies in choosing the right fundamentals and deciding when to share them between different target constants.

In this paper, we introduce a Constraint Programming (CP) model to solve MCM. It basically consists in a translation into arithmetic constraints of the relationship associating the operands of each adder to its output. We add an *Among* global constraint to verify that all target constants are produced, and a *Count* global constraint to check that adders with no children (*i.e.*, sink nodes in the adder graph) are associated with target constants. To minimize the number of adders, we solve a sequence of decision problems, aiming at deciding whether there exists a solution with x adders with x ranging from $\#T$ (because we need at least on adder by target constant) to the smallest value for which the answer is yes. This smallest value is denoted N.

To improve the solution process, we introduce symmetry breaking rules that specify how to break ties when different fundamentals may be used to produce a same target constant. These rules are ensured thanks to a new global constraint called *Minimal Fundamental Sequences* (MFS) and we introduce a polynomial-time propagation algorithm for this constraint. The basic idea of this algorithm is to incrementally detect when target constants can be obtained from the current set of fundamentals and instantiate first non-instantiated fundamentals to these values. It relies on the implementation of a Boolean function for deciding whether a fundamental can be obtained from two others. As this function is called thousands of times during the propagation, we introduce a new linear implementation instead of the quadratic implementation of the literature.

We show that our MFS propagator, when combined with a specific variable ordering heuristic, allows us to solve MCM in polynomial time whenever the optimal number of adders N is equal to

Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

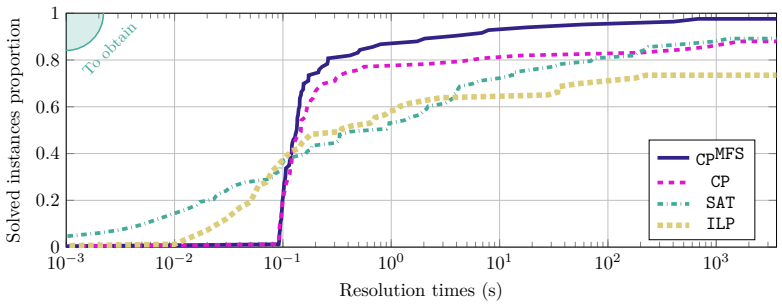

Figure 2 Evolution of the cumulative percentage of solved instances with respect to the run time

$\#T$. More generally, we show that our propagator naturally provides a Fixed Parameter Tractable (FPT) approach for MCM running in $\mathcal{O}(f(K, w) \cdot \#T^{\mathcal{O}(1)})$ time where $K := N - \#T$, w is the number of bits needed to encode the largest constant in T, and f is exponential. In other words, our solving approach is exponential in K and w, but not in the number of targets in T. To the best of our knowledge, this is the first FPT algorithm proposed for MCM.

We experimentally evaluate our approach on a widely used benchmark extracted from a collection of digital filter designs. We denote `CP` our CP model without the MFS constraint, and $\texttt{CP}^{\texttt{MFS}}$ our CP model with the MFS constraint. Both models are implemented in Java with Choco [3], and are compared with state-of-the-art approaches, *i.e.*, the ILP approach of [2], denoted `ILP`, and the SAT approach of [1], denoted `SAT`. We show that `CP` is competitive with `ILP` and `SAT`, and that $\texttt{CP}^{\texttt{MFS}}$ clearly outperforms all other existing approaches on the considered benchmark. More precisely, when looking at Fig. 2, we note that `ILP` and `SAT` are more successful for time limits smaller than 0.1s. Actually, Choco always spends around 0.1s to build the model, so that the total run time is nearly always larger than 0.1s. However, once the model is built, many instances are very quickly solved and $\texttt{CP}^{\texttt{MFS}}$ becomes much more successful than `SAT` and `ILP`.

Our new MFS global constraint, specifically designed for the MCM problem, may be generalized to a broader class of optimization problems known as Straight-Line Programs (SLP). While SLPs retain the notion that the production of fundamentals depend on the values of the previously producted ones, they abstract the operators used to compute reachability, including their arity. Consequently, further work will aim at providing an abstract propagator that users can specialize for their specific SLP.

2012 ACM Subject Classification Theory of computation → Constraint and logic programming

Keywords and phrases Constraint Programming, Multiple Constant Multiplication, Hardware Optimization

Digital Object Identifier 10.4230/LIPIcs...

References

1 Nicolai Fiege, Martin Kumm, and Peter Zipf. Bit-level optimized constant multiplication using boolean satisfiability. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 71(1):249–261, 2024.

2 Rémi Garcia and Anastasia Volkova. Toward the multiple constant multiplication at minimal hardware cost. *IEEE Transactions on Circuits and Systems I: Regular Papers*, pages 1–13, 2023.

3 Charles Prud'homme and Jean-Guillaume Fages. Choco-solver: A java library for constraint programming. *Journal of Open Source Software*, 7(78):4708, 2022.