Simultaneous Approximation of Constraint Satisfaction Problems

Amey Bhangale\textsuperscript{1(\S)}, Swastik Kopparty\textsuperscript{2}, and Sushant Sachdeva\textsuperscript{3}

\textsuperscript{1} Department of Computer Science, Rutgers University, New Brunswick, USA
\texttt{amey.bhangale@rutgers.edu}
\textsuperscript{2} Department of Mathematics and Department of Computer Science, Rutgers University, New Brunswick, USA
\texttt{swastik.kopparty@rutgers.edu}
\textsuperscript{3} Department of Computer Science, Yale University, New Haven, USA
\texttt{sachdeva@cs.yale.edu}

Abstract. Given $k$ collections of 2SAT clauses on the same set of variables $V$, can we find one assignment that satisfies a large fraction of clauses from each collection? We consider such simultaneous constraint satisfaction problems, and design the first nontrivial approximation algorithms in this context.

Our main result is that for every CSP $\mathcal{F}$, for $k < \tilde{O}(\log^{1/4} n)$, there is a polynomial time constant factor Pareto approximation algorithm for $k$ simultaneous Max-$\mathcal{F}$-CSP instances. Our methods are quite general, and we also use them to give an improved approximation factor for simultaneous Max-$w$-SAT (for $k < \tilde{O}(\log^{1/3} n)$). In contrast, for $k = \omega(\log n)$, no nonzero approximation factor for $k$ simultaneous Max-$\mathcal{F}$-CSP instances can be achieved in polynomial time (assuming the Exponential Time Hypothesis).

These problems are a natural meeting point for the theory of constraint satisfaction problems and multiobjective optimization. We also suggest a number of interesting directions for future research.

1 Introduction

The theory of approximation algorithms for constraint satisfaction problems (CSPs) is a very central and well developed part of modern theoretical computer science. Its study has involved fundamental theorems, ideas, and problems such as the PCP theorem, linear and semidefinite programming, randomized rounding, the Unique Games Conjecture, and deep connections between them [3, 4, 15, 21, 29, 30].
In this paper, we initiate the study of simultaneous approximation algorithms for constraint satisfaction problems. A typical such problem is the simultaneous Max-CUT problem: Given a collection of $k$ graphs $G_i = (V, E_i)$ on the same vertex set $V$, the problem is to find a single cut (i.e., a partition of $V$) so that in every $G_i$, a large fraction of the edges go across the cut.

More generally, let $F$ be a set of bounded-arity predicates on $[q]$-valued variables. Let $V$ be a set of $n$ $[q]$-valued variables. An $F$-CSP is a weighted collection $W$ of constraints on $V$, where each constraint is an application of a predicate from $F$ to some variables from $V$. For an assignment $f : V \to [q]$ and a $F$-CSP instance $W$, we let $\text{val}(f, W)$ denote the total weight of the constraints from $W$ satisfied by $f$. The Max-$F$-CSP problem is to find $f$ which maximizes $\text{val}(f, W)$. If $F$ is the set of all predicates on $[q]$ of arity $w$, then Max-$F$-CSP is also called Max-$w$-CSP.

We now describe the setting for the problem we consider: $k$-fold simultaneous Max-$F$-CSP. Let $W_1, \ldots, W_k$ be $F$-CSPs on $V$, each with total weight 1. Our high level goal is to find an assignment $f : V \to [q]$ for which $\text{val}(f, W_\ell)$ is large for all $\ell \in [k]$.

These problems fall naturally into the domain of multi-objective optimization: there is a common search space, and multiple objective functions on that space. Since even optimizing one of these objective functions could be NP-hard, it is natural to resort to approximation algorithms. Below, we formulate some of the approximation criteria that we will consider, in decreasing order of difficulty:

1. **Pareto approximation:** Suppose $(c_1, \ldots, c_k) \in [0, 1]^k$ is such that there is an assignment $f^*$ with $\text{val}(f^*, W_\ell) \geq c_\ell$ for each $\ell \in [k]$.
   An $\alpha$-Pareto approximation algorithm in this context is an algorithm, which when given $(c_1, \ldots, c_k)$ as input, finds an assignment $f$ such that $\text{val}(f, W_\ell) \geq \alpha \cdot c_\ell$, for each $\ell \in [k]$.

2. **Minimum approximation:** This is basically the Pareto approximation problem when $c_1 = c_2 = \ldots = c_k$. Define $\text{OPT}$ to be the maximum, over all assignments $f^*$, of $\min_{\ell \in [k]} \text{val}(f^*, W_\ell)$.
   An $\alpha$-minimum approximation algorithm in this context is an algorithm which finds an assignment $f$ such that $\min_{\ell \in [k]} \text{val}(f, W_\ell) \geq \alpha \cdot \text{OPT}$.

3. **Detecting Positivity:** This is a very special case of the above, where the goal is simply to determine whether there is an assignment $f$ which makes $\text{val}(f, W_\ell) > 0$ for all $\ell \in [k]$.

When $k = 1$, minimum approximation and Pareto approximation correspond to the classical Max-CSP approximation problems (which have received much attention). Our focus in this paper is on general $k$. It is useful to think of $k$ as $O(1)$, or a slowly growing function of $n$, say $\log \log n$. As we will see in the discussions below, the nature of the problem changes quite a bit for $k > 1$. In particular, direct applications of classical techniques like random assignments and convex programming relaxations fail to give even a constant factor approximation for values of $k$ greater than certain threshold.

The theory of exact multiobjective optimization has been very well studied, (see eg. [10,27] and the references therein). A common theme in this area is to