Gadgets, Approximation, and Linear Programming

[Errata]

Luca Trevisan   Gregory B. Sorkin

Madhu Sudan   David P. Williamson

We apologize for two errors in the FOCS proceedings version of our paper [2].

The first error was typographical. In the introductory example illustrating a reduction from 3SAT to MAX 2SAT, the 10 clauses replacing $C_k = X_1 \vee X_2 \vee X_3$ should be

$$X_1,\ \ X_2,\ \ X_3,\ \ \neg X_1 \vee \neg X_2,\ \ \neg X_2 \vee \neg X_3,\ \ \neg X_3 \vee \neg X_1$$

$$Y^k,\ \ X_1 \vee \neg Y^k,\ \ X_2 \vee \neg Y^k,\ \ X_3 \vee \neg Y^k.$$

The second error was in Lemma 3.1 and, as a consequence, Corollary 3.3. However, the further results of the paper all depend on the more specific Lemma 3.5, which is correct, so our principal claims are unaffected.

Lemma 3.1 claimed that for any gadget reducing a constraint f to a constraint family $\mathcal{F}$, there exists an equivalent gadget with at most K auxiliary variables, where $K = K_{f,\mathcal{F}}$ is a finite bound. The error was pointed out by Karloff and Zwick [1], who provide a counterexample in which no finite gadget achieves optimality.

Since the proof of Lemma 3.5 referred to that of Lemma 3.1, we give here a self-contained proof.

Lemma 3.5 If an α-gadget Γ reducing f to a hereditary family $\mathcal{F}$ has a witness function for which two auxiliary variables are identical (i.e. $b_{j'}(\cdot) \equiv b_j(\cdot)$), or if an auxiliary variable is identical to a primary variable ($b_{j'}(\vec{a}) \equiv a_j$) then there is an α'-gadget Γ' using one fewer auxiliary variable, and with $\alpha' \leq \alpha$. If Γ is strict, so is Γ'.

Proof: We define a new gadget Γ' obtained from Γ by replacing each occurrence of $X_{j'}$ by X_j and argue that Γ' is an α'-gadget reducing f to $\mathcal{F}$ for some $\alpha' \leq \alpha$.

For any constraint C of Γ, define $red(C)$ as follows. If $X_{j'}$ does not occur in C, then $red(C) = C$. Otherwise, we tentatively define $red(C)$ as the constraint obtained from C by replacing the occurrence of $X_{j'}$ by an occurrence of X_j. If C did not originally involve X_j, then $red(C)$ is a valid constraint from $\mathcal{F}$. If C did involve X_j already, then $red(C)$ contains two occurences of X_j, which is not allowed by our definition. However, the hereditary property of $\mathcal{F}$ yields either an equivalent constraint $C' \in \mathcal{F}$ or else the constant function 0 or 1. In this case we reset $red(C)$ to C' or the appropriate constant.

If $\Gamma = (C_1, \ldots, C_m, w_1, \ldots, w_m)$, then define a new gadget $\Gamma' = (red(C_1), \ldots, red(C_m), w_1, \ldots, w_m)$. Correspondingly, let $\vec{b}'(\vec{a})$ be identical to $b(\vec{a})$ but with $b_{j'}$ eliminated. Γ' has one fewer auxiliary variable ($X_{j'}$ never occurs in Γ').

By construction, $\Gamma'(\vec{a}, \vec{b}'(\vec{a})) \equiv \Gamma(\vec{a}, \vec{b}(\vec{a}))$, so Γ' satisfies the gadget-defining equation (2). (Similarly, for strict gadgets Γ, Γ' satisfies (4)). Also, the range of the universal quantification for Γ' is smaller than that for Γ, therefore Γ' satisfies inequalities (1) and (3). If constants are produced, subtracting them from both sides of the gadget-defining inequalities (1–4) produces an $(\alpha - w)$-gadget, where w is the total weight on clauses replaced by 1's. $\square$

Acknowledgements

We thank Howard Karloff and Uri Zwick for bringing both errors to our attention.

References

[1] H. Karloff and U. Zwick. Personal communication. September 1996.

[2] L. Trevisan, G.B. Sorkin, M. Sudan, and D.P. Williamson. Gadgets, approximation and linear programming. In *Proc. of the 37th Annual IEEE Symposium on Foundations of Computer Science*, 1996.