22.01.23

Theorem: The k-network flow update problem where every edge is used by at most three flow pairs is NP-hard for $k = 2, 10$.

Proof by reduction from 4-SAT.

The proof is based on the NP-hardness proof for $k = 6$ in [2].

Reduction

W.l.o.g. every variable occurs at least once positively and at least once negatively.

Given a 4CNF formula $\mathcal{C}$ with n variables $x_1, \dots, x_n$ and m clauses $C_1, \dots, C_m$, we construct the corresponding update flow network G as follows. First, we introduce a clause gadget for each clause and a variable gadget for each variable in a way that is independent of $\mathcal{C}$. Then, we connect the variable and clause gadgets in a way that depends on $\mathcal{C}$. Finally, we take the remaining steps necessary to ensure that G is indeed a feasible update flow network.

Clause gadgets

For every clause $C_i = (l_{i_1} \vee l_{i_2} \vee l_{i_3} \vee l_{i_4})$, we ~~define~~ construct the corresponding clause gadget C^i as follows.

I think I want to draw the analogy to the syntax tree for C_i and describe the clause gadget this way.

We introduce a *clause vertex* u^i which is used by three flow pairs L, R, B. The idea is to guarantee that clause C_i is satisfied iff ~~if~~ node $b(u^i, L)$ is updated before node $b(u^i, B)$ or node $b(u^i, R)$ is updated before $b(u^i, B)$, or, equivalently? in other words, $b(u^i, B)$ cannot be updated (b(u^i, R)) unless at least one of $b(u^i, L), b(u^i, R)$ has? have been updated before. Intuitively, if $b(u^i, L)$ is updated before $b(u^i, B)$, then the left half $(l_{i_1} \vee l_{i_2})$ of (Right half $(l_{i_3} \vee l_{i_4})$] of C_i is satisfied. Moreover, for literals $l_{i_1}, l_{i_2}, l_{i_3}, l_{i_4}$, we introduce *literal vertices* $u^i_1, u^i_2, u^i_3, u^i_4$. The idea is to guarantee that literal l_{i_4} ($l_{i_2}, l_{i_3}, l_{i_4}$ evaluates to 1 if node ~~$b(u^i_1, F(i_1)$ $b(u^i_2, F$~~ $b(u^i_1, F(i_1)]$ $(b(u^i_2, F(i_2)]$, $b(u^i_3, F(i_3)]$, $b(u^i_4, F(i_4)]]$ is updated before node $b(u^i, B)$, where flow pair $F(i_4)$ depends on whether literal l_{i_4} is positive or negative. The rest of clause gadget C^i connects ~~the~~ clause vertex u^i with literal vertices $u^i_1, u^i_2, u^i_3, u^i_4$. We introduce vertices $u^i_{3\ell}, u^i_{3r}$ which are used by three flow pairs $\tilde{L}, \tilde{R}, \tilde{B}$. Vertex $u^i_{3\ell}$ (u^i_{3r}) can be thought of the clause vertex for the left ~~leg~~ half $(l_{i_1} \vee l_{i_2})$ (right half $(l_{i_3} \vee l_{i_4})$] of clause C_i. Consequently, if $b(u^i_{3\ell}, \tilde{L})$ $(b(u^i_{3r}, \tilde{R})]$ is updated before $b(u^i_{3\ell}, \tilde{B})$, then the left half l_{i_1} (right half l_{i_2}] of $(l_{i_1} \vee l_{i_2})$ is satisfied. Finally, we introduce vertices $\tilde{u}^i_{3\ell}, \tilde{u}^i_{3r}$ to connect u^i with $u^i_{3\ell}, u^i_{3r}$, respectively. We remark that $\tilde{u}^i_{3\ell}, \tilde{u}^i_{3r}$ are not necessary for this proof (we could instead connect u^i with $u^i_{3\ell}$ via L and u^i with u^i_{3r} via R). They are necessary, however, for the proof of corollary ~.

Can we do without $F(i_4)$?

I think I don't want to mention $\tilde{u}^i_{3\ell}, \tilde{u}^i_{3r}$ in the intuition.

We proceed with the detailed description of clause gadget C^i.

Clause vertices. We introduce two vertices u^i, v^i and add edge $(u^i, v^i]$ to flows L°, R°, B^u. Similarly, we introduce vertices $u^i_{12}, v^i_{12}, u^i_{34}, v^i_{34}$ and add edges $(u^i_{12}, v^i_{12}], (u^i_{34}, v^i_{34}]$ to flows $\tilde{L}^\circ, \tilde{R}^\circ, \tilde{B}^u$.

Literal vertices. We introduce vertices $u^i_1, v^i_1, u^i_2, v^i_2, u^i_3, v^i_3, u^i_4, v^i_4$ and add edges $(u^i_1, v^i_1], (u^i_3, v^i_3]$ to $\tilde{L}^u$ and $(u^i_2, v^i_2], (u^i_4, v^i_4]$ to $\tilde{R}^u$.

Auxiliary vertices. We introduce vertices $\tilde{u}^i_{12}, \tilde{v}^i_{12}, \tilde{u}^i_{34}, \tilde{v}^i_{34}$ and add edge $(\tilde{u}^i_{12}, \tilde{v}^i_{12}]$ to $\tilde{L}^u$ and $\tilde{B}^\circ$ and $(\tilde{u}^i_{34}, \tilde{v}^i_{34}]$ to $\tilde{R}^u$ and $\tilde{B}^\circ$.

Finally, we add the following edges to connect the clause gadget:

- ... (see first draft)

Variable gadgets

For every variable x_i, we construct the corresponding variable gadget X^i as follows. We introduce a variable vertex x^i which is used by three flow pairs $X, \overline{X}, B$. The idea is to guarantee (1) if blocks $b(x^i, X]$ is updated before blocks $b(x^i, B]$, then variable x_i is assigned 1, (2) if blocks $b(x^i, \overline{X}]$ is updated before $b(x^i, B]$, then x_i is assigned 0, and (3) not both $b(x^i, X]$ and $b(x^i, \overline{X}]$ can be updated before $b(x^i, B]$.

We proceed with the detailed description of variable gadget X^i.

We introduce flow pairs $X, \overline{X}$ and vertices x^i, y^i as well as and add edge $(x^i, y^i]$ to $X^u, \overline{X}^u, B^\circ$. Moreover, we introduce vertices $x^i_0, y^i_0, x^i_1, y^i_1$ and add edges $(x^i_0, y^i_0]$ to $\overline{X}^\circ$ and $(x^i_1, y^i_1]$ to X°. To connect the variable gadgets, we add edges $(x^i, x^i_0], (y^i_0, y^i]$ to X° and $(x^i, x^i_1], (y^i_1, y^i]$ to X°. We remark that $x^i_0, y^i_0, x^i_1, y^i_1$ are not necessary for this proof. They are, however, necessary, however, for the proof of Corollary ~.

Connecting the variable and clause gadgets

We now connect every variable gadget X^i to a every clause gadget C^i if variable x_i occurs in clause C_i. For every j and every (B_j) (x^i_j) all literal vertices

More precisely, we introduce flow pairs B_0, B_1 such that B_0 connects vertex x^i_0 to ~~or a clause gadget corresponding to literal vertex u^i_1 has u^i_2, u^i_3, u^i_4 if $\ell_{i_1} = \overline{x_i}$ ... $(\ell_{i_3} = \overline{x_i})$~~. We remark again that B_0, B_1 are not necessary for this proof (we could instead connect directly x^i directly to u^i_j via $\overline{X}$ $(X]$ if $(\ell_{i_1} = \overline{x_i}$ $(\ell_{i_j} = x_i)]$, but they are necessary for the proof of Corollary ~. More formally, let $P_i = \{p^i_1, \dots, p^i_{c_i}\}$ be the set of indices of the clauses containing literal x_i and $\overline{P}_i = \{\overline{p}^i_1, \dots, \overline{p}^i_{c_i}\}$ be the set of indices of the clauses containing literal $\overline{x_i}$. Moreover, let $\pi(i,j]$ denote the position of literal x_i in clause C_j and $\overline{\pi}(i,j]$ denote the position of literal $\overline{x_i}$ in C_j. We add the following edges:

- ... (see first draft)

27.01.23

Completing the feasible update flow network

We introduce vertices s, t and create an (s,t)-path for all flows by adding the following edges:

- ... (see first draft)

We defined the flow pairs $X, \bar{X}, L, R, \bar{L}, \bar{R}, B, B_0, B_1, \tilde{B}$ and set all demands to 1.

Edge capacities are defined as follows:

- We set the capacity to 2 for edges $(x^i, y^i), (u^i, v^i), (u^i_{x2}, v^i_{x2}), (u^i_{3\bar{c}}, v^i_{3\bar{c}})$
- 1 $(x^i_0, y^i_0), (x^i_1, y^i_1), (\tilde{u}^i_{x2}, \tilde{v}^i_{x2}), (\tilde{u}^i_{3c}, \tilde{v}^i_{3c}), (u^i_1, v^i_1), (u^i_2, v^i_2), (u^i_3, v^i_3), (u^i_4, v^i_4)$
- All remaining edge capacities are set to 10, that is, the number of flows. Hence they cannot violate any capacity constraints.

~~TODO Verify feasibility. Use that we assumed that every variable occurs at least once positively and at least once negatively.~~

Let us quickly verify that G is indeed a feasible update flow network. To verify that every flow is indeed a (s,t)-path, see Table ~. Recall that we assumed that every variable x_i occurs at least once positively and at least once negatively. Hence $P_i \neq \{\}$ and $\bar{P}_i \neq \{\}$. Thus B_0 and B_1 form (s,t)-paths. To verify that all capacity constraints are satisfied for both the old flow network and the update flow network, see Table ~. All edges not listed have capacity 10, that is, the number of flows, ~~and hence cannot cause violate the corresponding~~. Hence the corresponding capacity constraints cannot be violated.

Table with (s,t)-path for every flow.

TODO loop-freedom