

# Sorting Networks - SAT cu Green Filter

Alexandru Lungu

October 2020

## 1 Introducere

Sorting networks este o problema computationala ce presupune constructia unei multimi ordonate  $C_n = \{(i, j) | 1 \leq i \leq j \leq n\}$ , astfel incat pentru orice input  $o$ , numar natural cu  $n$  biti, reseaua bazata pe multimea ordonata  $C_n$  va genera doar solutii din multimea  $Sol_n$ , ce contine doar numerele naturale cu  $n$  biti sortati ( $|Sol_n| = n + 1$ ). Evident, se doreste a se minimaliza  $k = |C_n|$ . Aceasta definitie a fost formulata avand in vedere zero-one principle.

Reteaua aferenta unei multimi ordonate  $C_n$  presupune un set de operatii peste numarul natural input  $o_0$ . La pasul 1, se va considera primul element al multimii  $C_n$ , iar bitii de pe pozitiile  $(i_1, j_1)$  vor fi sortati, optinandu-se astfel  $o_1$ . La pasul  $k$ , se va considera  $o_{k-1}$  peste care se vor sorta bitii de pe pozitiile  $(i_k, j_k)$  pentru a obtine  $o_k$ . Solutia retelei este  $o_{|C_n|}$ . Reteaua este valida daca  $\forall o_0 \in Input_n, o_{|C_n|} \in Sol_n$ , unde  $Input_n$  este multimea tuturor numerelor naturale cu  $n$  biti.

## 2 Model - encodare SAT

Modelul pe care l-am abordat a fost bazat pe o encodare SAT (2). Aceasta presupune a gasi o asignare pentru multimea de variabile boolene  $g_{i,j,k}$ , ce reprezinta alegerea unui comparator intre bit-ul  $i$  si  $j$  la pasul  $k$ . In paralel, vom avea nevoie de variabilele boolene  $o_m^k$ , ce reprezinta validitatea unei configuratii  $m$  la pasul  $k$ . Avand aceste variabile putem adauga urmatoarele constrangeri:

- Pentru fiecare  $k$  (adica nivel de alegere) va trebui sa avem un singur comparator. Prin urmare, pentru nivelul  $k'$ , un singur  $g_{i,j,k'}$  ar trebui sa fie adevarat.
- Propagarea validitatii  $o_m^k$  in functie de comparatorii alesi trebuie sa fie modelata astfel. Daca pentru un nivel  $k'$ , avem  $g_{i,j,k'}$  adevarat, atunci  $o_m^k$  va trebui sa fie adevarat doar in cazul in care exista un eventual  $o_w^{k-1}$  adevarat (unde  $w$  este o configuratie premergatoare configuratiei  $m$ ).
- Toate  $o_m^0$  trebuie sa fie adevarate. Asta inseamna ca reseaua trebuie sa fie valida pentru orice input.

- Toate  $o_m^k$ , unde  $m \notin Sol_n$  trebuie sa fie false

Aceasta abordare poarta denumirea de "Fixed-size forward encoding (SFWD)"  
(2)

### 3 Model - Green Filter

In continuarea implementarii modelului bazat pe encodarea SAT, am adaugat niste clauze bazate pe Green Filter (1). Asadar, modelul nu garanteaza completitudinea atat timp cat acest filtru este folosit, insa imbunatateste semnificativ viteza de evaluare.

### 4 Rezultate

Rezultatele sunt construite astfel incat sa se remarce diferenta de timp produsa de adaugarea Green Filter. De asemenea este important de vizualizat numarul de clauze si variabile pentru a estima potentialul unei astfel de abordari. Pentru implementare am folosit frameworkul Z3 (3) pentru evaluarea formulei logice generate la final. Timpii sunt calculati in secunde in formatul "timp pentru generarea formulei" + "timp pentru rezolvarea formulei".

n	k	VARIABLE COUNT	STANDARD TIME	GREEN-FILTER TIME
3	3	68	0.25 + 0.00	0.25 + 0.00
4	5	192	0.32 + 0.00	0.32 + 0.00
5	9	570	0.69 + 0.02	0.67 + 0.02
6	12	1300	2.00 + 0.65	1.87 + 0.08
7	16	3009	6.37 + 36.03	6.23 + 0.31
8	19	6400		19.11 + 0.92
9	25	15418		61.65 + 15.04

Se observa o diferenta remarcabila intre timpul de generare si timpul de rezolvare. Acest aspect este datorat principal limbajului folosit (Python 3), care nu este suficient de performant in generarea de clauze pentru formula. Putem considera totusi ca timpul de generare ar trebui sa fie mai putin important, incat generarea poate fi optimizata in mai multe moduri (programare distributiva, limbaj suport mai eficient etc.).

Timpul rezolvarii formulei este mai relevant, incat acest proces este partea activa a algoritmului ce incearca a gasi combinatia de conectori potriviti unei eventuale retele valide.

### 5 Concluzii si posibile optimizari

Rezultatele obtinute aici sunt comparabile cu cele obtinute in (2), insa optimizarea cu Green-Filter aduce un nou ordin de magnitudine in aceasta abor-

dare. Deși rezultatele nu sunt comparabile cu cele din (1), există optimizări în cadrul formulei:

- Eliminarea variabilelor ce sunt asignate direct din definiția constructivă a formulei ( $o_m^0$  de exemplu poate fi înlocuit cu True pentru a simplifica formula).
- Eliminarea variabilelor din primele  $gf$  nivele, unde  $gf$  este numărul de conectori impusi de Green-Filter.
- Implementarea unui SBCK (2), ce obține rezultate mai bune din punct de vedere a timpului. Pentru confirmarea existenței unei rețele  $C_n, |C_n| = k$  știind  $n$  și  $k$ .

## References

- [1] Cristian Frăsinaru, Mădălina Răschip *Greedy Best-First Search for the Optimal-Size Sorting Network Problem*. Faculty of Computer Science, "Alexandru Ioan Cuza" University, Iasi, Romania
- [2] Jose A. R. Fonollosa *SAT encodings for sorting networks, single-exception sorting networks epsilon-halvers*. Department of Signal Theory and Communications, Universitat Politècnica de Catalunya, Barcelona, Spain
- [3] Leonardo de Moura and Nikolaj Bjørner *Z3: An Efficient SMT Solver*. Microsoft Research, One Microsoft Way, Redmond, WA, 98074, USA