

Maciej DROZDOWSKI
Politechnika Poznańska

INFORMACYJNY ROZMIAR PROBLEMÓW KOMBINATORYCZNYCH

Streszczenie. Problemy kombinatoryczne są relacjami łańcuchowymi i jako takie mogą być mierzone ilością zawartej informacji. Na przykład problem spełnialności wyrażeń boolowskich zawiera wykładniczą ilość informacji w funkcji rozmiaru instancji. Jednocześnie informacyjna złożoność Kołmogorowa tego problemu jest stała. Problem spełnialności może też być rozwiązany w wielomianowym czasie o ile dysponujemy wykładniczą ilością informacji.

ON INFORMATION SIZE OF COMBINATORIAL PROBLEMS

Summary. Combinatorial problems are string relations and as such can be assessed with the amount of comprised information. For example, boolean satisfiability comprises exponential amount of information. Simultaneously Kolmogorov information complexity of this problem is constant. Moreover, satisfiability can be solved in polynomial time if exponential amount of information is available.

1. Wprowadzenie

Zdolność rozwiązywania trudnych problemów optymalizacji kombinatorycznej wiąże się z ilością dostępnej informacji. Na przykład, w heurystykach populacyjnych duży rozmiar i różnorodność populacji pozwalają uzyskiwać rozwiązania dobrej jakości. Dla algorytmów zachłannych istnieją przykłady, których te algorytmy nie rozwiązują do optymalności. Tak jest i w przypadku wierzchołkowego kolorowania grafu. Dla *deterministycznych* algorytmów zachłannych podano grafy, których te metody nie kolorują optymalnie [9]. Z drugiej strony, algorytm kolorujący wierzchołki w losowej kolejności nie ma tzw. grafu trudnego do kolorowania i nie może deterministycznie upaść. Algorytm ten ma dostęp do nieograniczonej ilości informacji w postaci generatora liczb losowych. Twierdzenia o braku darmowych obiadów [12], analizy zbieżności algorytmów symulowanego wyżarzania [5], analizy związku między rozwiązywalnością problemu spełnialności a strukturalną entropią powiązań zmiennych [14], odwołują się do kwestii ilości informacji. Liczne klasy złożoności obliczeniowej oraz zależności między nimi zostały sformułowane dla probabilistycznych wersji maszyn Turinga [1], lub probabilistycznie weryfikowanych dowodów [3], przy dostępności dodatkowej informacji w postaci losowych łańcuchów.

W tej pracy przyjmuje się, że problemy kombinatoryczne są obiektami, dla których można zmierzyć zawartość informacji. Przedmiotem analizy nie są konkretne cechy, zależności, czy struktury ale ilość informacji. Wykazane zostanie że ilość informacji w problemie spełnialności wyrażeń boolowskich (dalej w skrócie SAT) rośnie wykład-

niczo z rozmiarem instancji, co stoi w sprzeczności ze stałą informacyjną złożonością Kołmogorowa tego samego problemu. Ponadto, wykazane zostanie że gdyby było możliwe dostarczenie wykładniczej ilości informacji algorytmowi rozwiązującemu problem SAT, to problem ten można by rozwiązać w wielomianowym czasie.

Dalsza organizacja pracy jest następująca: Pojęcia i definicje zostaną przedstawione w następnym rozdziale. Rozdział 3. zawiera dowód że problem spełnialności ma wykładniczą ilość informacji, a z drugiej strony informacyjna złożoność Kołmogorowa jest stała. W rozdziale 4. pokazane zostanie, że problem SAT potencjalnie mógłby być rozwiązany w wielomianowym czasie przy wykładniczej ilości dostępnej informacji. Ostatni rozdział zawiera konkluzje i propozycje dalszych badań.

2. Definicje

Problem spełnialności wyrażeń boolowskich (SAT) jako problem przeszukiwania jest zdefiniowany następująco:

Definicja 1. Spełnialność wyrażeń boolowskich

Dane: klauzule (tzn. sumy) $k_j, j = 1, \dots, m$, zmiennych binarnych, lub ich negacji, wybranych z n -elementowego zbioru $\{x_1, \dots, x_n\}$.

Polecenie: Wyznacz przydział wartości 0/1 do zmiennych $x_1, \dots, x_n$, dla którego funkcja $F(I, \bar{x}) = \prod_{j=1}^m k_j$ przyjmuje wartość 1. Jeżeli taki przydział nie istnieje, to zgłoś $\emptyset$.

Dane wejściowe to inaczej instancja I problemu SAT. Rozmiar instancji I , tj. długość łańcucha kodującego dane I zapisane zgodnie z pewną rozsądną regułą kodowania e [6], będziemy oznaczać $|I|$. Instancję dla której istnieje wektor $\bar{x}$ spełniający warunek $F(I, \bar{x}) = 1$ będziemy nazywać instancją *pozytywną*, w przeciwnym przypadku będziemy mówić że instancja jest *negatywna*. Dla prostoty argumentacji będziemy przyjmować, że I jest zakodowana jako binarny łańcuch, a bity są jednostkami ilości informacji.

Niech Σ^+ oznacza zbiór łańcuchów kodujących instancje i ich rozwiązania dla pewnego problemu przeszukiwania, np. SAT, wykorzystując do tego alfabet znaków Σ i regułę kodowania e . Problemy przeszukiwania są binarnymi relacjami łańcuchowymi (ang. string relation) zdefiniowanymi następująco [6]:

Definicja 2. Problem przeszukiwania Π jest relacją łańcuchową

$$R[\Pi, e] = \left\{ (a, b) : \begin{array}{l} a \in \Sigma^+ \text{ koduje instancję } I \in D_\Pi \text{ zgodnie z regułą} \\ e, \text{ natomiast } b \in \Sigma^+ \text{ jest rozwiązaniem } s \in S_\Pi(I) \\ \text{zakodowanym zgodnie z regułą } e \end{array} \right\},$$

gdzie D_Π jest dziedziną, czyli zbiorem wszystkich instancji, problemu Π , $S_\Pi(I)$ jest zbiorem rozwiązań dla instancji I problemu Π .

Definicja 3. Informacyjny rozmiar problemu jest liczbą bitów potrzebną aby go zapisać jako binarną relację łańcuchową.

Sens powyższej definicji zasługuje na obszerniejsze wyjaśnienie. Problemy przeszukiwania są binarnymi relacjami łańcuchowymi $R[\Pi, e]$ w sensie Definicji 2. (por.

[6]). $R[\Pi, e]$ jest zbiorem par (a, b) łańcuchów, w których a koduje instancję, a b rozwiązanie. Informacyjny rozmiar problemu Π to liczba bitów zapisu zbioru par łańcuchów (a, b) tworzących $R[\Pi, e]$. Definicja 3. odwołuje się do szeroko używanych pojęć takich jak: relacja binarna, relacja łańcuchowa, pary (a, b) łańcuchów zapisanych z użyciem skończonego alfabetu Σ , rozsądna reguła kodowania e . Informacyjny rozmiar problemu można więc starać się szacować od góry i od dołu jako liczbę bitów wystarczającą i niezbędną do zapisania $R[\Pi, e]$. Definicja 3. różni się od definicji złożoności informacyjnej wg Kołmogorowa [8].

Definicja 4. *Dla pewnego binarnego łańcucha X , złożoność Kołmogorowa, to długość najkrótszego algorytmu który pozwala odtworzyć łańcuch X .*

Informacyjny rozmiar problemu (Definicja 3.) różni się od złożoności informacyjnej Kołmogorowa (Definicja 4.) tym, że informacyjny rozmiar problemu dotyczy relacji łańcuchowej już odtworzonej przez hipotetyczny algorytm z Definicji 4. Zauważmy, że algorytmy rozwiązujące problemy, np. kombinatoryczne, muszą podawać rozwiązanie dla każdej instancji, czyli dla każdej możliwej pary (a, b) łańcuchów w relacji łańcuchowej, algorytm musi określić łańcuch b dysponując łańcuchem a . Z takiego punktu widzenia, algorytm rozwiązujący jakiś problem musi być w stanie odtworzyć relację łańcuchową w pełnym informacyjnym rozmiarze problemu (wg Definicji 3.).

Definicja 5. *Prawdziwie Losowy Ciąg Bitów (PLCB) jest sekwencją bitów która nie ma krótszej reprezentacji.*

Chociaż losowość jest powszechnie wykorzystywanym pojęciem, to wyznaczenie informacyjnej złożoności Kołmogorowa, tzn. długości najkrótszego programu który może odtworzyć zadany łańcuch znaków, jest problemem nierozstrzygalnym. Dlatego PLCB jest pojęciem pierwotnym, tak jak zbiór w teorii mnogości, lub punkt w geometrii. Dla celów niniejszej pracy istotny jest fakt że PLCB nie może zostać skrócony (np. skompresowany) i aby reprezentować PLCB o długości N bitów potrzebny jest łańcuch o długości N bitów.

3. Ilość informacji w problemie spełnialności

Twierdzenie 6. *Informacyjny rozmiar problemu spełnialności wyrażeń boolowskich rośnie co najmniej wykładniczo z rozmiarem instancji.*

Dowód. Dowód składa się z trzech kroków: (1) konstrukcji wykładniczej liczby instancji, (2) wstawienia PLCB do zbioru tych instancji, (3) przeliczenia ilości informacji z zależności od liczby zmiennych boolowskich n do zależności od rozmiaru instancji $|I|$.

(1) Przyjmijmy, że danych jest $n \geq 3$ zmiennych boolowskich, dla których sformułujemy $4n$ klauzul: $k_{i1} = x_a + x_b + \bar{x}_i$, $k_{i2} = \bar{x}_a + x_b + \bar{x}_i$, $k_{i3} = x_a + \bar{x}_b + \bar{x}_i$, $k_{i4} = \bar{x}_a + \bar{x}_b + \bar{x}_i$ dla $i = 1, \dots, n$, gdzie $\bar{x}_i$ oznacza zmienną x_i , która zostanie zapisana z negacją lub bez negacji (to zostanie objaśnione w dalszym tekście). Zmienne x_a, x_b należy wybrać tak by $a \neq b$ oraz $a, b \neq i$. Dla $n \geq 3$ istnieje więc $(n-1)(n-2)/2$ możliwych par a, b dla każdego i . Żaden przydział wartości do zmiennych x_a, x_b nie pozwala na to aby klauzule $k_{i1}, k_{i2}, k_{i3}, k_{i4}$ jednocześnie miały wartość 1. Iloczyn tych klauzul

może stać się jedynką tylko gdy $\widetilde{x}_i = 1$ we wszystkich czterech klauzulach. Spełnienie formuły $F = k_{11}k_{12}k_{13}k_{14} \dots k_{n4}$ zależy więc od wartości $\widetilde{x}_i$ dla $i = 1, \dots, n$. Tylko jeżeli wartości $\widetilde{x}_i$ zapiszemy konsekwentnie jako x_i albo $\overline{x}_i$ w każdej klauzuli $k_{i1}, k_{i2}, k_{i3}, k_{i4}$, to F może stać się jedynką. To daje możliwość skonstruowania 2^n różnych pozytywnych instancji o 2^n różnych rozwiązaniach.

(2) Przyjmijmy, że dany jest prawdziwie losowy ciąg 2^n bitów, tj. PLCB o długości 2^n . Niech $j \in \{1, \dots, 2^n\}$ oznacza jeden z bitów PLCB. Wartość j można zapisać binarnie, wówczas $j[i]$ dla $i = 1, \dots, n$ będzie i -tym bitem kodu binarnego j . Jeśli wartość j -tego bitu PLCB jest równa 1, to zmienne $\widetilde{x}_i$ należy ustawić tak, że wartość j zakodowana binarnie będzie spełniać $k_{1i}, \dots, k_{4i}$. Na przykład, jeżeli j -ty bit PLCB jest jedynką i $j[i] = 1$, to $\widetilde{x}_i$ należy zapisać jako x_i . Jeżeli j -ty bit PLCB jest jedynką i $j[i] = 0$, to $\widetilde{x}_i$ należy zapisać jako $\overline{x}_i$. Tak więc, dla bitu o wartości 1 zapisanego na j -tej pozycji PLCB, j -ta instancja zakodowana w powyższy sposób jest pozytywna. Jeżeli jednak j -ty bit PLCB ma wartość zero, to przynajmniej jedna ze zmiennych x_i w odpowiadających jej klauzulach $k_{1i}, \dots, k_{4i}$ jest zapisana w różny sposób – przynajmniej raz z negacją i przynajmniej raz bez negacji. O ile więc j -ty bit PLCB ma wartość zero, to j -ta instancja zbudowana w powyższy sposób jest negatywna. W ten sposób PLCB o długości 2^n został ukryty w problemie 3-SAT i jego informacyjny rozmiar jest co najmniej $\Omega(2^n)$.

(3) Pozostaje jeszcze przedstawić oszacowanie ilości informacji w SAT jako funkcję rozmiaru instancji $|I|$, a nie liczby zmiennych n . Przyjmując tzw. jednorodne kryterium kosztu [2] żadna rozważana liczba nie jest większa niż pewna stała K . Długość łańcucha kodującego instancje w powyższej konstrukcji wynosi $|I| = 4n \times 3 \log K + \log K = 12n \log K + \log K$ gdyż wystarczy zakodować liczbę zmiennych n , każda zmienna generuje cztery klauzule $k_{1i}, \dots, k_{4i}$, w których indeksy trzech zmiennych zapisane są na $(3 \log K)$ bitach. Negacja zmiennej, lub brak negacji, zapisuje się w jednym z $\log K$ bitów kodujących zmienną. Liczba bitów PLCB (a więc i informacyjny rozmiar problemu spełnialności) wynosi $2^n = 2^{(|I| - \log K)/(12 \log K)} = 2^{|I|/(12 \log K)} 2^{-1/12}$, czyli jest $\Omega(2^{d_1 |I|})$, gdzie $d_1 = 1/(12 \log K) > 0$.

Przyjmując tzw. logarytmiczne kryterium kosztu [2], liczba bitów potrzebnych do zapisania n wynosi $\lfloor \log n \rfloor + 1$, a aby zapisać indeks zmiennej binarnej i jej negację lub brak negacji, potrzeba $\lfloor \log n \rfloor + 2$ bitów. Długość łańcucha kodującego instancje skonstruowane wg przedstawionego powyżej sposobu wynosi $|I| = 12n(\lfloor \log n \rfloor + 2) + \lfloor \log n \rfloor + 1 \leq 15n \log n = dn \ln n$, gdzie $n > 2^{24}$ oraz $d = 15/\ln 2 \approx 21.6404$. Funkcja odwrotną do $y = (cx \ln x)$ dla $c > 0$, jest $\frac{y}{c} / W(\frac{y}{c})$, gdzie W jest tzw. funkcją W Lamberta [11]. Dla dużych y funkcję W można przybliżyć jako $W(y) = \ln y - \ln \ln y + O(1)$. Dla zadanej długości instancji $|I|$ otrzymujemy $n \geq \frac{|I|}{d} / W(\frac{|I|}{d}) \approx \frac{|I|}{d} / (\ln \frac{|I|}{d} - \ln \ln \frac{|I|}{d} + O(1)) \geq \frac{|I|}{d} / (2 \ln \frac{|I|}{d}) \geq \frac{|I|}{d} / (2 \ln |I| - 2 \ln d) \geq |I| / (2d \ln |I|)$, dla dużych $|I|$. Długość PLCB zakodowanego w problemie spełnialności wynosi $2^n \geq 2^{|I|/(d_2 \ln |I|)}$ gdzie $d_2 = 2d$. Zauważmy, że $2^{|I|/(d_2 \ln |I|)}$ dla wystarczająco dużych $|I|$ przekracza co do wartości każdą wielomianową funkcję $O(|I|^\ell)$, gdyż $\ln(|I|^\ell) < |I|/(d_2 \ln |I|)$, gdzie ℓ jest stopniem wielomianu. $\square$

Zauważmy, że konstrukcja w powyższym dowodzie pozwala w prosty sposób odczytać bit łańcucha PLCB z instancji. Nie zmienia to jednak faktu, że sam łańcuch PLCB nie ma krótszej reprezentacji i nie zmniejsza to ilości informacji w problemie SAT. Pod-

sumowując, problem spełnialności zawiera PLCB o rozmiarze co najmniej $\Omega(2^{d_1|I|})$ dla kryterium jednorodnego, lub co najmniej $\Omega(2^{|I|/(d_2 \ln |I|)})$ dla kryterium logarytmicznego. Powyższe twierdzenia o wykładniczej ilości losowej informacji w problemie spełnialności mają ważne konsekwencje: (1) Losowość (reprezentowana przez PLCB) i złożoność (reprezentowana przez problem SAT) w pewnych sytuacjach mogą być nierozróżnialne (podobnych obserwacji dokonano w odniesieniu do trudności generowania liczb pseudolosowych [10, 13]). (2) O ile uczenie maszynowe polega na rozpoznawaniu wzorców w danych, co służy konstrukcji algorytmów inferencji, to problem spełnialności jest niewyuczalny bo zawiera w sobie PLCB który nie ma żadnych wzorców. To spostrzeżenie można rozszerzyć na wszystkie problemy **NP**-zupełne gdyż istnieją między nimi transformacje wielomianowe.

Problem SAT można także scharakteryzować określając jego złożoność Kołmogorowa. Chociaż w ogólności złożoność Kołmogorowa jest nieobliczalna, to dla SAT jest ona stała. Skoro SAT to relacja łańcuchowa $R[\Pi, e]$, to należy podać stałej długości program, który odtworzy $R[\Pi, e]$. Taki program przedstawia następujący pseudokod:

def G :

 for n in $1, 2, \dots$:

 for s in łańcuchy_binarne_długości($2n + 2n2^{2n}$) :

$I = n_2|^\sim s$; InstniejeRozwiązanie=False

 for $\bar{x}$ in łańcuchy_binarne_długości(n) :

 if $V(I, \bar{x})$:

 zapisz($I, \bar{x}$)

 InstniejeRozwiązanie=True

 break

 if not InstniejeRozwiązanie :

 zapisz($I, \emptyset$)

Program odtwarzający problem spełnialności jako binarną relację łańcuchową składa się z generatora G który wylicza wszystkie instancje I , ich potencjalne rozwiązania $\bar{x}$ oraz weryfikatora V który sprawdza czy $F(I, \bar{x}) = 1$. Taki weryfikator istnieje gdyż spełnialność jest problemem z klasy **NP**. Dla pewnej liczby zmiennych binarnych n , można wyliczyć wszystkie zakodowane binarnie instancje I oraz n -bitowe łańcuchy rozwiązań tak jak wylicza się kolejne liczby całkowite. W powyższym zapisie nazwa łańcuchy_binarne_długości(ℓ) reprezentuje wszystkie liczby zapisane binarnie na dokładnie ℓ bitach. Symbol $|$ oznacza konkatencję łańcuchów, n_2 oznacza binarnie zapisaną wartość n , dla prostoty prezentacji nie-binarny znak $\sim$ rozdziela binarny zapis wartości n od binarnego zapisu ciągu klauzul s . Łańcuch s koduje kolejno: na $2n$ bitach liczbę klauzul o długości $2n$ bitów, a na kolejnych $2n2^{2n}$ bitach same klauzule k_j . Jeżeli klauzul jest mniej niż 2^{2n} to nadmiarowe bity w s są ignorowane. Wyliczanie kolejnych łańcuchów s i $\bar{x}$ oznacza konieczność ustawicznego powiększania o jeden liczb zapisanych binarnie w łańcuchach dowolnej długości przez algorytm o stałym rozmiarze kodu. Jest to możliwe przy zserializowanym algorytmie dodawania. Szczegóły takiej metody realizującej powyższy pseudokod przedstawiono w [4].

Spostrzeżenie 7. *SAT ma stałą złożoność Kołmogorowa.*

Zwróćmy uwagę na rozbieżność Twierdzenia 6. i Spostrzeżenia 7. Z jednej strony spełnialność zawiera wykładniczą ilość informacji, a z drugiej ma stałą złożoność

informacyjną. Ta sytuacja jest ilustracją paradoksu. Z jednej strony złożoność informacyjna Kołmogorowa zbioru *wszystkich* możliwych łańcuchów generowanych przez powyższy program jest $O(1)$, a przy ustalonym rozmiarze instancji wynosi ona $O(\log |I|)$, gdyż trzeba zapisać ograniczenie rozmiaru łańcuchów I . Z drugiej strony zbiór *wszystkich* możliwych łańcuchów generowanych przez powyższy program ma losowy podzbiór o informacyjnej złożoności Kołmogorowa $\Omega(2^{|I|})$, co pokazuje Twierdzenie 6. Tę paradoksalną sytuację można przedstawić jako dwuosobową grę między algorytmem rozwiązującym SAT i adwersarzem, w której to grze adwersarz ze zbioru wszystkich możliwych łańcuchów o rozmiarze $|I|$ potrafi wyciągnąć instancje o informacyjnej złożoności Kołmogorowa $\Omega(2^{|I|})$. Kolejny paradoks polega na tym, że chociaż algorytm gra z adwersarzem o sile wyrażanej zdolnością do wytworzenia łańcuchów o złożoności Kołmogorowa $\Omega(2^{|I|})$, to sam algorytm dysponuje tylko informacją o jednej instancji o rozmiarze $|I|$ oraz informacją, którą sam w sobie zawiera. W takiej dwuosobowej grze algorytm i adwersarz odtwarzają SAT jako relację łańcuchową. Powyższą rozbieżność między Twierdzeniem 6. i Spostrzeżeniem 7. można także postrzegać z perspektywy faktu, że informacyjna złożoność Kołmogorowa nie uwzględnia liczby kroków niezbędnych do odtworzenia zadanego łańcucha X hipotetycznym algorytmem z Definicji 4.

Informacyjny rozmiar problemu spełnialności (wg Definicji 3.) można także oszacować od góry. W parze łańcuchów $(a, b) \in R[\Pi, e]$, a koduje instancję o rozmiarze $|I|$. Jeżeli a koduje instancję pozytywną to łańcuch b musi kodować n -bitowe rozwiązanie $\bar{x}$. Jeżeli a koduje instancję negatywną to łańcuch b musi kodować symbol $\emptyset$ o stałej długości. Jeżeli łańcuch a nie koduje poprawnie instancji problemu spełnialności to będziemy go traktować jak instancję negatywną. Tak więc, aby zapisać dowolną parę łańcuchów (a, b) wystarczy $O(|I|)$ bitów. Przy kodowaniu binarnym łańcucha a o długości $|I|$ istnieje co najwyżej $2^{|I|}$ różnych instancji o zadanej tu długości. Wystarczy więc $O(|I|2^{|I|})$ bitów do zakodowania problemu spełnialności jako binarnej relacji łańcuchowej $R[\text{SAT}, e]$.

Spostrzeżenie 8. *Informacyjny rozmiar problemu spełnialności jest $O(|I|2^{|I|})$.*

4. Złożoność problemu spełnialności przy wykładniczej ilości informacji

W tym rozdziale analizujemy jaki wpływ na rozwiązywalność problemu spełnialności mogłoby mieć dysponowanie dużą ilością informacji. Przyjmijmy, że platforma obliczeniowa pozwala na *polilogarytmiczny* czas dostępu do informacji. Oznaczać to będzie, że czas odnalezienia informacji w zbiorze o rozmiarze Z jest $O(p(\log(Z)))$, gdzie p jest wielomianem. Przyjęcie że czas dostępu do danych jest polilogarytmiczny jest praktycznie uzasadnione, gdyż w przypadku danych przechowywanych w pamięci operacyjnej (tzw. pamięć RAM) czas dostępu jest stały, a w przypadku danych przechowywanych na mediach o dostępie sekwencyjnym (np. pliki w urządzeniach zewnętrznych) czas dostępu jest $O(\log(Z))$ dzięki stosowaniu różnego rodzaju struktur indeksujących, np. B/B+ drzew. Problem spełnialności potencjalnie można rozwiązać przez odczytywanie wstępnie obliczonych rozwiązań z pamięci. Opisuje to poniższe spostrzeżenie.

Spostrzeżenie 9. *Problem spełnialności można rozwiązać w wielomianowym czasie na systemach z polilogarytmicznym czasem dostępu używając $O(|I|2^{|I|})$ informacji odwołując się do uprzednio obliczonych rozwiązań.*

Dowód. Dla zadanej instancji I problemu SAT, powyższe stwierdzenie realizuje pseudokod:

rozwiązanie $\leftarrow$ TablicaRozwiązań[I];

Rozwiązanie jest tu odczytywane z TablicyRozwiązań, a instancja I jest indeksem określającym pozycję w tablicy. Ponieważ tablica ta przechowuje $O(2^{|I|})$ elementów $\bar{x}, \emptyset$, każdy o rozmiarze co najwyżej $O(|I|)$, cała tablica ma rozmiar $O(|I|2^{|I|})$. Przy polilogarytmicznym czasie dostępu rozwiązanie może zostać odnalezione w wielomianowym czasie $O(p(|I| + \log |I|))$. $\square$

Chociaż idea przedstawiona w Spostrzeżeniu 9. nie skaluje się do dowolnych rozmiarów problemu i w tym sensie nie jest praktyczna, to wskazuje że ilość informacji którą niesie ze sobą sam algorytm, może rzutować na złożoność obliczeniową. Podobny efekt uzyskuje się dla niejednorodnych klas złożoności z wykładniczą podpowiedzią (ang. non-uniform complexity) [7]. Od ukazania się publikacji [7] technologia rozwiązywania problemów optymalizacji kombinatorycznej istotnie rozwinęła się i znanych jest wiele odmiennych metod dostarczania informacji, inaczej podpowiedzi, do algorytmów rozwiązujących takie problemy.

5. Podsumowanie

W niniejszej pracy wykazano, że problem spełnialności wyrażeń boolowskich, postrzegany jako binarna relacja łańcuchowa, zawiera ilość informacji rosnącą wykładniczo wraz z rozmiarem instancji. Z drugiej strony informacyjna złożoność Kołmogorowa dla tego samego problemu jest stała. Jednocześnie problem spełnialności jest rozwiązywalny w wielomianowym czasie jeśli możliwe jest manipulowanie wykładniczymi ilościami informacji. Może to oznaczać, że teoria złożoności informacyjnej (Kołmogorowa) nie uwzględnia ilości informacji zdobywanej przez algorytm w trakcie jego wykonywania. Intrygującym wyzwaniem teoretycznym byłoby wykazanie, że ilość informacji którą zawiera SAT jest $\Theta(|I|2^{|I|})$. Praktycznym wyzwaniem jest zbadanie jaka jest skalowalność algorytmów które nie miałyby w sobie stałej ilości informacji. Naturalnymi kandydatami wydają się metody uczenia maszynowego, które na różne sposoby rozpoznają wzorce w danych i uzyskane tą drogą informacje wykorzystują do konstrukcji algorytmów inferencji rozwiązujących nowe nieznane wcześniej instancje. Jest to tym bardziej intrygujące, że Twierdzenie 6. implikuje że problemy obliczeniowo trudne, takie jak SAT, są niewyuczalne.

LITERATURA

1. Adleman L.: Two theorems on random polynomial time. In 19th Annual Symposium on Foundations of Computer Science (sfcs 1978), Ann Arbor MI USA, 1978, p.75–83.
2. Aho A., Hopcroft J.E., Ullman J.D.: The Design and Analysis of Computer Algorithms. Addison-Wesley Publishing Company, Reading MA, 1974.
3. Arora S., Lund C., Motwani R., Sudan M., Szegedy M.: Proof verification and the hardness of approximation problems. J. ACM, 1998, 45(3), p.501–555.

4. Drozdowski M.: On SAT information content, its polynomial-time solvability and fixed code algorithms. *arXiv:2401.00947*, 2024.
5. Fleischer M., Jacobson S.H.: Information theory and the finite-time behavior of the simulated annealing algorithm: Experimental results. *INFORMS Journal on Computing*, 1999, 11(1), p.35–43.
6. Garey M.R., Johnson D.S.: *Computers and Intractability: A guide to the theory of NP-completeness*. Freeman, San Francisco, 1979.
7. Karp R.M., Lipton R.J.: Some connections between nonuniform and uniform complexity classes. In *Proceedings of the twelfth annual ACM symposium on theory of computing (STOC'80)*, Association for Computing Machinery, Los Angeles CA USA, 1980, p. 302–309,
8. Kolmogorov A.N.: Three approaches to the quantitative definition of information. *Problemy Peredachi Informatsii*, 1965, 1(1), p.3–11. (tłumaczenie z rosyjskiego).
9. Kubale M. (ed.): *Optymalizacja dyskretna, modele i metody kolorowania grafów*. Wydawnictwa Naukowo Techniczne, Warszawa, 2002.
10. Nisan N., Wigderson A.: Hardness vs randomness. *J. Comput. Syst. Sci.*, 1994, 49(2), p.149–167.
11. Weisstein E.W.: Lambert W-function. *MathWorld—A Wolfram Web Resource*. accessed Aug 05, 2025.
12. Wolpert D.H., Macready W.G.: No free lunch theorems for optimization. *IEEE Trans. on Evolutionary Computation*, 1997, 1(1), p.67–82.
13. Yao A.C.: Theory and application of trapdoor functions. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, Chicago IL USA, Nov. 1982, p. 80–91.
14. Zhang Z., Xu D., Zhou J.: A structural entropy measurement principle of propositional formulas in conjunctive normal form. *Entropy*, 2021, 23(3), art. No. 303.