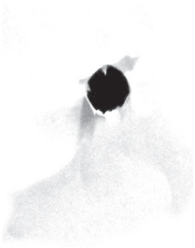

Warto na koniec dodać, że charakter kwazi-cząstek można kontrolować przez zmianę wymiaru przestrzeni, w której je badamy. Jeśli zmusimy elektrony i dziury do ruchu tylko w jednej płaszczyźnie, np. wytwarzając bardzo cienką, kilkunanometrową warstwę kryształu, wtedy mamy do czynienia z kwazi-wszechświatem dwuwymiarowym (tzw. studnią kwantową), jeśli tę warstwę pokroimy jak makaron, dostaniemy obiekty jednowymiarowe (druty kwantowe), a jak dalej posiekamy ten makaron na drobne kawałeczki, dostaniemy kropki kwantowe – obiekty zerowymiarowe (kwazi-zerowymiarowe), które można traktować jak sztuczne atomy. Zastosowania? Np. lasery półprzewodnikowe to studnie kwantowe, w których dokonujemy skuteczniej anihilacji elektronów i dziur; kropki kwantowe były niedawno hitem nowoczesnych luminoforów w telewizorach.

Nasze możliwości są w zasadzie nieograniczone. Badaniami tego typu zjawisk zajmuje się fizyka ciała stałego i nanotechnologia, zastosowania sięgają urządzeń półprzewodnikowych, optycznych, przetwarzania informacji – także kwantowej, bo komputery kwantowe też można zaprojektować z kwazi-cząstek. Badania trwają, przecież jest tyle kwazi-wszechświatów do odkrycia!

Jak proste problemy stały się trudne

*Michał WŁODARCZYK**

* doktorant, Instytut Informatyki, Wydział Matematyki, Informatyki i Mechaniki, Uniwersytet Warszawski

Dawno, dawno temu, wierzono, że fundamentalne zasady rządzące światem są proste. Kiedy dziedzina nauki, zwana obecnie informatyką, dopiero raczkowała, naukowcy byli przekonani, że dla każdego problemu obliczeniowego można znaleźć efektywny algorytm, o ile poświęci się na to wystarczająco dużo czasu, kredy oraz kawy. Pojęcie „efektywnego algorytmu” oznaczało początkowo algorytm o czasie działania proporcjonalnym do rozmiaru danych wejściowych, ale i algorytmy o złożoności obliczeniowej $\mathcal{O}(n^2)$ czy $\mathcal{O}(n^3)$ były do zaakceptowania.

Pojawiły się jednak problemy, których pomimo wielu lat badań i hektolitrów wypitej kawy nikt nie umiał rozwiązać lepiej, niż generując wszystkie potencjalne rozwiązania i wybierając najlepsze z nich. Czas działania takich algorytmów jest zazwyczaj wykładniczy, zatem dla dużych danych działają one zdecydowanie wolniej niż te o złożoności wielomianowej. Do trudnych problemów należały:

- Pokrycie Wierzchołkowe: mając dany graf, znajdź najmniejszy podzbiór wierzchołków, stykający się z każdą krawędzią.
- Cykl Hamiltona: mając dany graf, rozstrzygnij, czy istnieje cykl odwiedzający każdy wierzchołek dokładnie raz.
- Spełnialność Formuł (SAT): mając daną formułę logiczną, złożoną ze zmiennych przyjmujących wartości prawda/fałsz oraz operatorów AND, OR, NOT, rozstrzygnij, czy można przypisać zmiennym takie wartości, aby formuła była prawdziwa.

Naukowcy zaczęli rozumieć to zjawisko lepiej na początku lat 70. Choć nie udało się efektywnie rozwiązać ani jednego z „trudnych” problemów, ani też wykluczyć istnienia algorytmów wielomianowych, udowodniono, że prawie wszystkie znane trudne problemy są równoważne na mocy wielomianowych redukcji pomiędzy nimi. Powstała teoria NP-trudności wyjaśniała, że efektywne rozwiązanie dowolnego z nich pociąga istnienie szybkich algorytmów dla pozostałych. Zamiast zastanawiać się nad setkami różnych problemów niezależnie, naukowcy zrozumieli, że „jądro” trudności jest takie samo. Choć dalej nie wiemy, czy da się je rozwiązać wielomianowo, całe zagadnienie sprowadza się do jednego kluczowego pytania: „czy P $\neq$ NP?”. Świat znów stał się prosty.

Co prawda, istnieje kilka problemów, o których do dziś nie wiemy, ani że są rozwiązywalne wielomianowo, ani że są NP-trudne, ale są to „niedobitki”.

Teoria NP-trudności okazuje się jednak czasem zbyt „grubo ciosana”, chociażby dla uczestników Olimpiady Informatycznej. Powiedzmy, że na zawodach pojawia się zadanie „rozstrzygnij, czy w danym zbiorze liczb naturalnych istnieją takie a, b, c, że $a + b = c$”. Najprostszy algorytm przegląda wszystkie trójki liczb i działa w czasie $\mathcal{O}(n^3)$. Znajomość podstawowych struktur danych pozwala na poprawę czasu działania do $\mathcal{O}(n^2 \log(n))$. Ale czy da się poprawiać dalej, np. do $\mathcal{O}(n \log(n))$? Inny problem: „mając dane dwa ciągi liczb, znajdź ich najdłuższy

W przypadku cyklu Hamiltona odpowiedź jest pozytywna i najlepszy znany algorytm działa w czasie $\mathcal{O}(1{,}66^n)$.

wspólny podciąg (niekoniecznie spójny)". Programowanie dynamiczne pozwala rozwiązać go w czasie $\mathcal{O}(n^2)$, ale skąd wiadomo, że nie da się lepiej?

Pierwszy zdefiniowany wyżej problem nazywa się 3SUM i, jako że ma rozwiązanie wielomianowe, teoria NP-trudności nie mówi nam nic o ograniczeniach dolnych na złożoność algorytmu. 3SUM okazuje się istotny w geometrii obliczeniowej i w latach 90. zastanawiano się, czy da się go rozwiązać w czasie np. $O(n^{1{,}99})$. Jako że zapasy kawy przeznaczonej na badania zaczynały się kurczyć, postawiono nową hipotezę, że nie istnieje algorytm podkwadratowy dla 3SUM. Z hipotezy wynikało, że nie istnieją podkwadratowe algorytmy dla wielu innych zagadnień, np. obliczania sumarycznej powierzchni n figur na płaszczyźnie. Jednak w przeciwieństwie do hipotezy P $\neq$ NP nieznane były żadne szokujące implikacje odrzucenia hipotezy 3SUM, toteż nie zyskała ona podobnego zainteresowania.

W międzyczasie rozwijała się teoria algorytmów wykładniczych. Postawiono wiele pytań typu „dla Cyklu Hamiltona znamy algorytm o złożoności $\mathcal{O}(2^n n^c)$, ale czy można to poprawić do $\mathcal{O}(1{,}99^n n^c)$?" Aby wykluczyć istnienie szybszych algorytmów wykładniczych, potrzebowano jeszcze silniejszych hipotez niż P $\neq$ NP, ale takich, których obalenie byłoby wielkim szokiem dla nauki – inaczej trudno przekonać innych do wiary w nową hipotezę. Jedną z nich jest *Strong Exponential Time Hypothesis* (SETH), która w uproszczeniu głosi, że problemu SAT nie tylko nie da się rozwiązać wielomianowo, ale nawet złożoność $\mathcal{O}(1{,}99^n n^c)$ jest nieosiągalna. W oparciu o tę hipotezę (lub jej słabszą wersję) uzasadniono, że różne znane algorytmy wykładnicze są optymalne i dopóki nie zrozumiemy lepiej problemu SAT, nie warto już nad nimi pracować. Zaczęliśmy widzieć więcej, a świat się znowu trochę uprościł.

Pisząc o „krótkich" wektorach, mamy na myśli długość ich zapisu lub, innymi słowy, wymiar przestrzeni, w której są zanurzone.

Wróćmy do świata wielomianowego i spójrzmy na następujący problem *Orthogonal Vectors*: „mając dane n (krótkich) wektorów, rozstrzygnij, czy istnieje wśród nich para wektorów prostopadłych (czyli o iloczynie skalarnym 0)". Oczywiście, łatwo go rozwiązać, sprawdzając wszystkie pary wektorów i – jak zapewne Czytelnik już się domyśla – nie wiemy, czy da się to zrobić sprytniej. W roku 2005 opublikowano zaskakującą redukcję, z której wynika, że algorytm dla Orthogonal Vectors o złożoności $\mathcal{O}(n^{1{,}99})$ pociągałby algorytm dla SAT o złożoności $\mathcal{O}(1{,}995^n n^c)$. Teorie złożoności wielomianowej i wykładniczej zaczęły się przenikać. Następnie pokazano redukcję z problemu Orthogonal Vectors do wspomnianego wcześniej problemu najdłuższego podciągu. Oznacza to, że kwadratowy algorytm obliczania najdłuższego podciągu (znany od lat 60. i wykorzystywany intensywnie w biologii obliczeniowej) jest optymalny przy założeniu SETH.

Badacze zachęceni tymi odkryciami doszli do wniosku, że pół wieku po narodzinach teorii NP-trudności informatyka jest już „gotowa", by stworzyć teorię dokładnej złożoności wielomianowej. Pojawiły się kolejne redukcje z problemów Orthogonal Vectors i 3SUM. W porównaniu do klasycznej teorii złożoności „krajobraz" redukcji okazał się bardziej skomplikowany: zamiast jednej hipotezy mamy ich kilka (choć niektóre są powiązane), a zamiast równoważności pomiędzy problemami często znamy jedynie implikacje w jedną stronę. Pewne miejsca w tym krajobrazie udało się już zrozumieć lepiej. Na przykład, dla każdego z poniższych problemów znany jest algorytm o złożoności $\mathcal{O}(n^3)$, a znalezienie algorytmu o złożoności $\mathcal{O}(n^{2{,}99})$ pociągałoby postęp dla wszystkich pozostałych:

- Obliczanie najkrótszych ścieżek pomiędzy wszystkimi parami wierzchołków w grafie.
- Szukanie w grafie trójkąta o ujemnej sumie wag krawędzi.
- Rozstrzyganie, czy dana funkcja zadaje metrykę.
- Obliczanie promienia grafu (takiego najmniejszego k, że istnieje wierzchołek v, z którego odległość do wszystkich innych jest nie większa niż k).

Czy faktycznie teoria dokładnej złożoności jest bardziej skomplikowana niż teoria NP-trudności, czy też najbliższe lata przyniosą kolejny przełom i kolejne uproszczenie świata? To pytanie jest obecnie obiektem intensywnych badań, m.in. na Uniwersytecie Warszawskim. Czytelników zainteresowanych naszym stanem wiedzy w tej dziedzinie odsyłam do artykułu Virginii V. Williams *Hardness of Easy Problems: Basing Hardness on Popular Conjectures such as the Strong Exponential Time Hypothesis*.

Rozwiązanie zadania M 1584.
Ciąg dany rekurencyjnie przez

$$a_0 = 2, \quad a_n = a_{n-1}^2 - n, \text{ dla } n \geqslant 1.$$

spełnia dla naturalnego n warunek $a_n > n$, co można sprawdzić indukcyjnie: dla 1, 2 i 3 sprawdzamy bezpośrednio, ponadto jeśli $k > 3$, to mamy $a_{k+1} = a_k^2 - (k+1) = k^2 - k - 1 > k + 1$. Kolejno otrzymujemy zatem

$$a_0 > 0,$$
$$a_1 = a_0^2 - 1 > 0,$$
$$a_2 = (a_0^2 - 1)^2 - 2 > 0,$$
$$a_3 = ((a_0^2 - 1)^2 - 2)^2 - 3 > 0,$$
$$a_4 = (((a_0^2 - 1)^2 - 2)^2 - 3)^2 - 4 > 0,$$
$$\vdots$$

Rozwiązując te nierówności względem a_0, otrzymujemy kolejno

$$a_0 > 1,$$
$$a_0 > \sqrt{1+\sqrt{2}},$$
$$a_0 > \sqrt{1+\sqrt{2+\sqrt{3}}},$$
$$\vdots$$
$$a_0 = 2 > \sqrt{1+\sqrt{2+\sqrt{3+\sqrt{\cdots+\sqrt{n}}}}}.$$