



Maximum Diversity Picking

Jakub Adamczyk • Maciej Kmąk • Hubert Miklas • Michał Szymocha • Maciej Wiśniewski

Opiekun: dr inż. Wojciech Czech

Wydział Informatyki, Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

Projektowanie nowych leków

- Eksperymenty w laboratorium są drogie i trudne
- Używamy metod obliczeniowych **AI/ML** oraz ***in silico*** [1]
- Chcemy wybrać **różnorodny podzbiór molekuł** do testów, pokrywając przy tym przestrzeń chemiczną
- Minimalizuje ryzyko eksperymentalne i koszty [1, 2]
- Skala to **setki milionów cząsteczek**, więc potrzebujemy skalowalności

Uczenie maszynowe i inne dziedziny

- **NLP**: coreset selection w treningu modeli LLM [3]
- **Computer vision**: redukcja duplikatów w zbiorach obrazów i wideo [4]
- **Systemy rekomendacyjne**: większa różnorodność sugestii dla użytkowników [5]
- **Active learning**: przyspieszenie zbieżności treningu przez zróżnicowanie populacji [6]

Skala problemu w praktyce

Rozmiar danych

$10^8 - 10^9$

wielkość współczesnych
bibliotek cząsteczek chemicznych

Złożoność problemu

NP-trudny

dokładne rozwiązanie
niewykonalne dla dużych N

Złożoność pamięciowa

$O(N^2)$

pamięć naiwnej macierzy
odległości - zbyt kosztowne

Problem:

Mając zbiór X o rozmiarze N i metrykę d , wybrać podzbiór S o rozmiarze K tak, żeby jego elementy były jak najbardziej różnorodne według wybranej metryki [1, 2, 3]

$$\max_{\substack{S \subseteq X \\ |S|=K}} \min_{\substack{x_i, x_j \in S \\ i \neq j}} d(x_i, x_j)$$

Dlaczego wybraliśmy wariant MaxMin?

- unika wybierania wielu podobnych punktów z jednego klastra [3, 4]
- nie faworyzuje pojedynczych ekstremalnych outlierów tak jak warianty MaxSum i MaxMean [4]
- równomiernie pokrywa przestrzeń danych [3, 5]
- działa dla dowolnych danych z metryką, np. molekuły, teksty, obrazy [6, 7, 8]

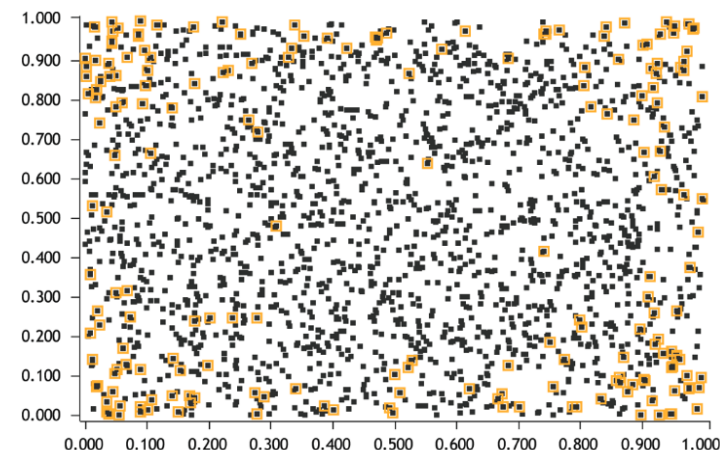
Konsekwencje praktyczne:

MaxSum maksymalizuje sumę odległości między parami – faworyzuje przez to obserwacje **odstające, mało reprezentatywne**

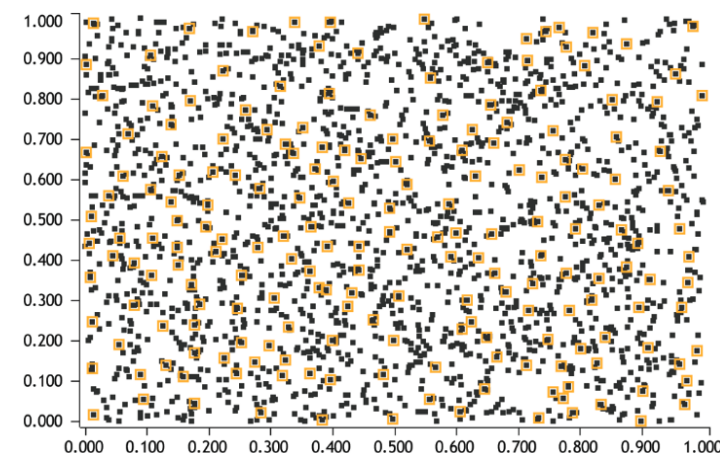
MaxMin maksymalizuje najmniejszą odległość między parami – podzbiór jest bardziej **równomiernie rozproszony**

Porównanie wariantów MaxSum i MaxMin [9]

MaxSum



MaxMin



1 Zoptymalizowany solver ILP z warm startem [1]

2 Leniwa implementacja algorytmu Ashtona [2, 3]

3 Algorytm Ashtona z optymalizacją Gonzaleza [2, 4]

4 Rozwiązanie randomizowane

5 iSIM [5]

6 GRASP [6]

7 Nevergrad [7]

8 Algorytmy genetyczne [8]

9 Symulowane wyżarzanie [9]

10 Cluster-then-pick [10]

Dokładna ewaluacja MaxMin ma złożoność $O(K^2)$ – niewykonalne dla dużych zbiorów
Zaproponowaliśmy 3 skalowalne ewaluatory aproksymacyjne

Losowe próbkowanie

Losujemy podzbiór par – szybkie i proste, ale mało dokładne

Extreme Value Theory (EVT) [11, 12]

Estymacja minimum rozkładu odległości, zgodnie ze wzorami teorii EVT

Approximate Nearest Neighbors (ANN) [13]

Nieparametryczne przeszukiwanie przestrzeni z indeksem HNSW

Zbiory małej skali (z rozwiązaniem dokładnym)

1 CHEMBL218_EC50 (N = 1 031)

2 BACE (N = 1 513)

3 CHEMBL244_Ki (N = 3 097)

4 CHEMBL233_Ki (N = 3 142)

5 CHEMBL214_Ki (N = 3 317)

6 CHEMBL234_Ki (N = 3 657)

Zbiory dużej skali

7 Tox21 (N = 7 831)

8 BindingDB (próbkowane N = 10 000)

9 COCONUT (próbkowane N = 10 000)

10 ChEMBL (próbkowane N = 10 000)

11 MOSES (próbkowane N = 10 000)

12 MUV (próbkowane N = 10 000)

13 PCBA (próbkowane N = 10 000)

14 ZINC250K (próbkowane N = 10 000)

Pytanie, na które odpowiadamy

Jaka metoda daje **największą różnorodność** dla określonego **budżetu czasowego**?

Rozwiązanie ma być użyteczne w zastosowaniach praktycznych

Ustandaryzowany protokół

- Wektory: fingerprinty molekularne ECFP4 (binarne, 2048 wymiarów)
- Metryka: odległość Tanimoto
- Dwa tryby: wybór dokładnie K oraz tryb progowy (wybór do określonej wartości MaxMin)
- Dokładna walidacja cząstek na każdym etapie
- Wiele ziaren losowych (seed) dla pewności statystycznej

Mierzone wartości

Jakość wartość MaxMin, jakość względna wobec optimum lub najlepszego znanego rozwiązania

Czas osobno przetwarzanie wstępne, sam wybór i całkowity czas wykonania

Scenariusze testowe

Małe N, małe K

$N \leq 5$ tys. · K 0,5–5%

wszystkie solvery, ILP jako rozwiązanie dokładne

Małe N, duże K

$N \leq 5$ tys. · K 10–25%

wszystkie solvery, ILP jako rozwiązanie dokładne

Duże N, małe K

N 0,005–10 mln · K 0,5–5%

standardowy przypadek: mały zestaw do badań

Duże N, duże K

N 0,005–10 mln · K 10–25%

badanie skalowalności solverów

Bardzo duże zbiory danych

$N \geq 10$ mln

tylko modyfikacje algorytmu Ashtona

Wybieranie z progiem podobieństwa

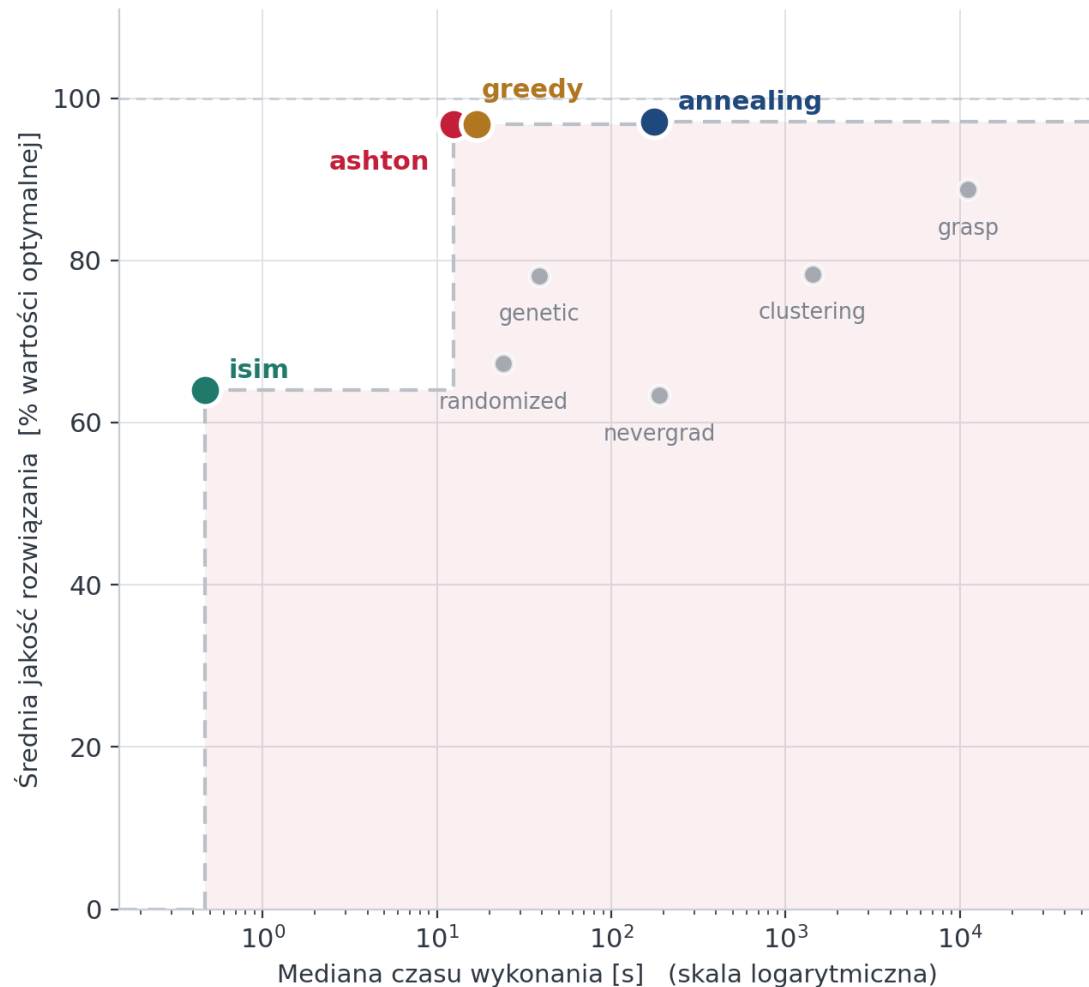
próg 0,45–0,90

ile cząsteczek zostaje wybranych przy określonej minimalnej różnorodności

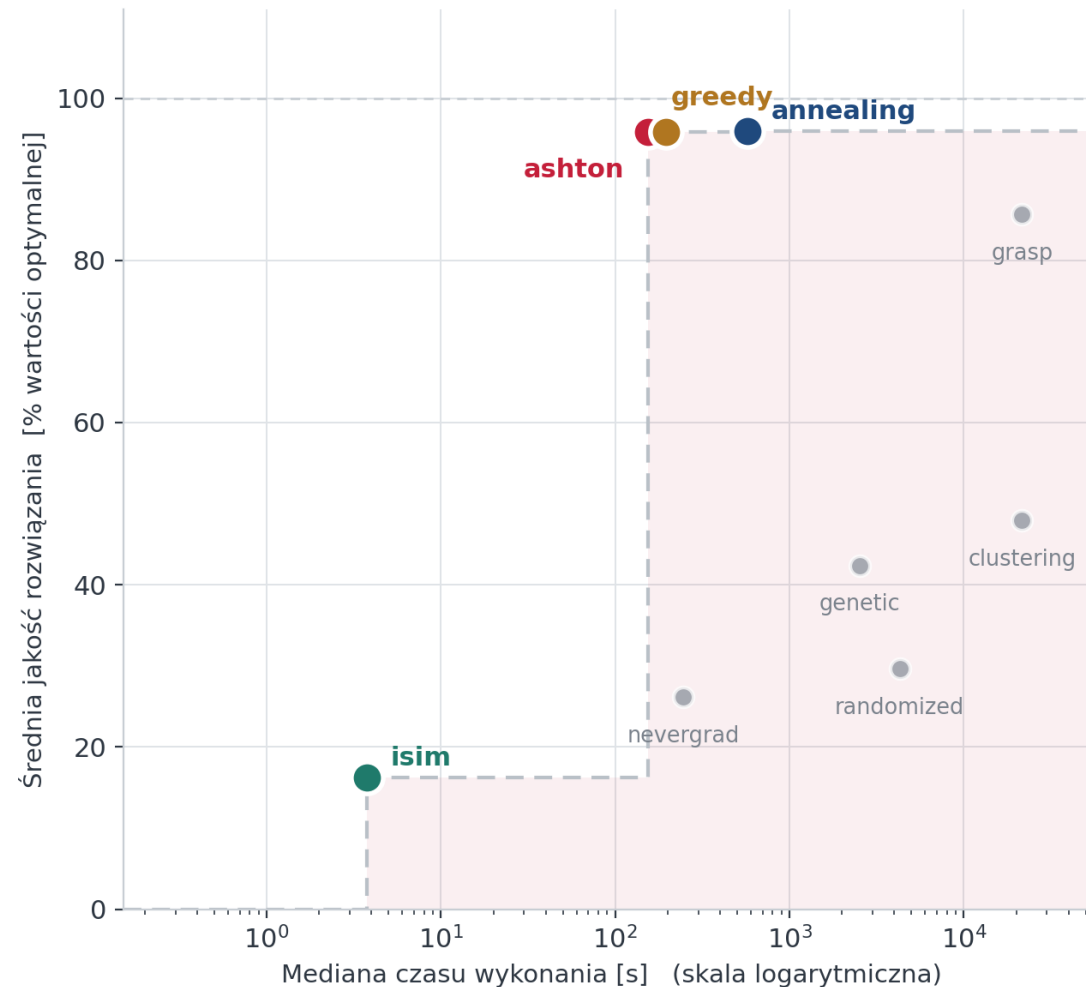
Gdy solver nie zakończy działania w określonym budżecie czasowym, zwraca najlepsze uzyskane rozwiązanie do tego momentu

Wybór najlepszego solvera w zależności od budżetu czasowego

Małe K (K = 0,5 - 5 % N)



Duże K (K = 10 - 25 % N)

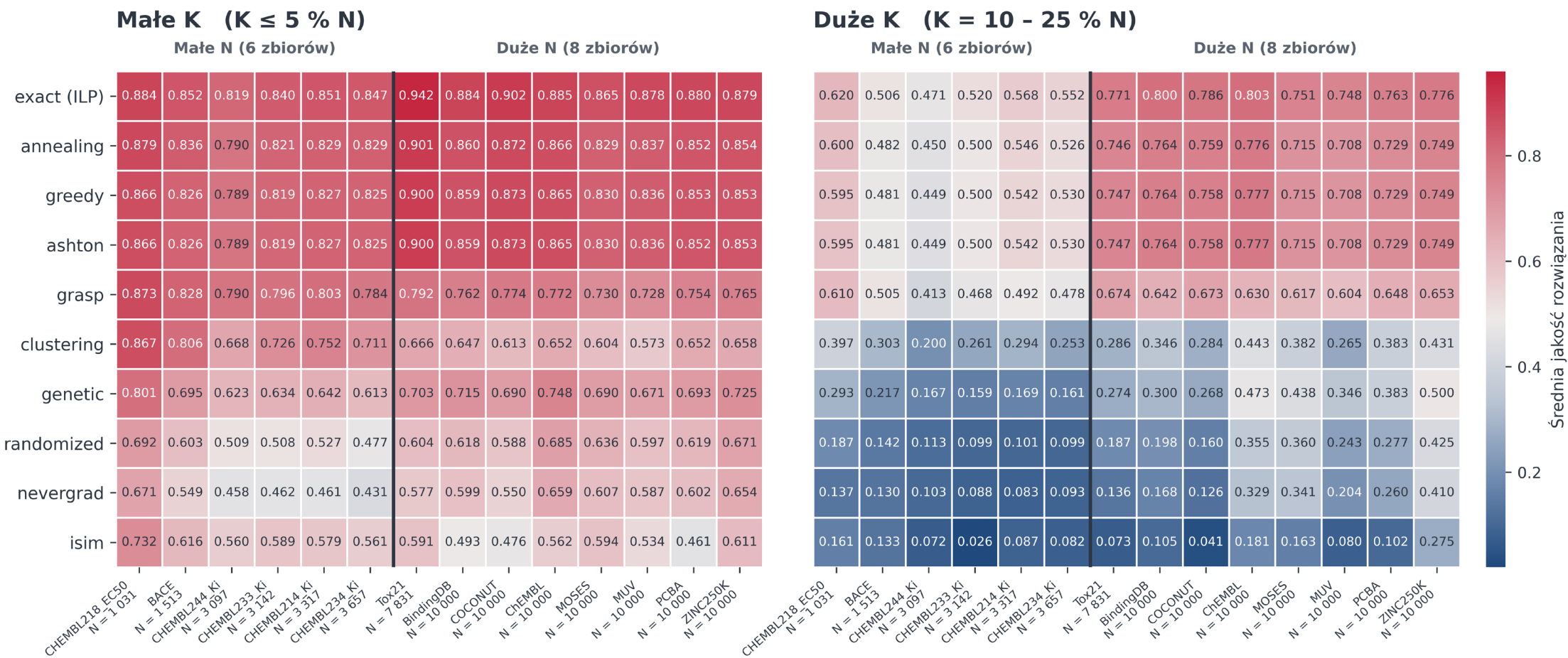


Porównanie jakości solverów

Małe K (k = 0,5 - 5 % N)										śr. %
exact (ILP) -		100	100	100	100	100	100	100	100	100
annealing -	0		63	63	90	100	92	100	100	79
ashton -	0	24		2	90	100	91	100	100	67
greedy -	0	24	2		90	100	91	100	100	67
grasp -	0	10	10	10		98	84	100	98	57
genetic -	0	0	0	0	1		52	99	85	37
clustering -	0	7	9	9	12	48		73	86	36
randomized -	0	0	0	0	0	0	26		48	17
isim -	0	0	0	0	2	14	13	52		16
nevergrad -	0	0	0	0	0	0	18	20	40	9
	exact	annealing	ashton	greedy	grasp	genetic	clustering	randomized	isim	nevergrad

Duże K (k = 10 - 25 % N)										śr. %
	100	100	100	100	100	100	100	100	100	100
0		45	45	100	100	100	100	100	100	77
0	43		2	99	100	100	100	100	100	72
0	42	1		99	100	100	100	100	100	71
0	0	1	1		100	100	100	100	100	56
0	0	0	0	0		38	100	100	100	38
0	0	0	0	0	62		93	99	95	39
0	0	0	0	0	0	7		90	76	19
0	0	0	0	0	0	0	8		18	3
0	0	0	0	0	0	5	21	81		12
	exact	annealing	ashton	greedy	grasp	genetic	clustering	randomized	isim	nevergrad

Jakość rozwiązania na poszczególnych zbiorach



Nasz wkład

- **10 solverów MaxMin MDP w jednym API** - jedna biblioteka, wszystkie typowe metody, od dokładnego ILP po heurystyki dla setek milionów elementów
- **Zoptymalizowany algorytm Ashtona** - równoległa, szybka implementacja, pamięć $O(N+K)$ zamiast $O(N^2)$
- **3 przybliżone ewaluatory jakości** pozwalające skalować metody do dużych wartości K

Co pokazały benchmarki

- **Proste heurystyki wygrywają** - metoda Ashtona pokonuje metaheurystyki jak GRASP, algorytmy genetyczne i Nevergrad
- **Stabilna hierarchia solverów** - analogiczne wyniki dla różnych K i zbiorów danych
- **Algorytm Ashtona** niemal zawsze jest **najlepszym** wyborem czasowym oraz pamięciowym

Kierunki dalszych prac

- Przeniesienie części obliczeń na GPU (CUDA)
- Benchmarki na zbiorach rzędu miliarda cząsteczek (np. ZINC22)
- Przygotowanie i publikacja artykułów naukowych

Podsumowanie wyników

Dobór maksymalnie różnorodnego podzbioru z biblioteki liczącej setki milionów cząsteczek **skraca się z godzin do minut**, zachowując jakość rozwiązania

256 mln

cząsteczek w jednym
przebiegu

19 min

zamiast około
3 godzin

$O(N + K)$

wymaganej pamięci
zamiast $O(N^2)$

Dostępność i zastosowania

- **Licencja MIT** – do zastosowań naukowych oraz komercyjnych
- **Wygodny i ogólny kod** – do zastosowań m.in. w chemioinformatyce, NLP, computer vision, systemach rekomendacyjnych
- **Ustandaryzowany benchmarking** – wygodne testowanie i wybór solverów w zależności od zadania i budżetu

Dziękujemy za uwagę!