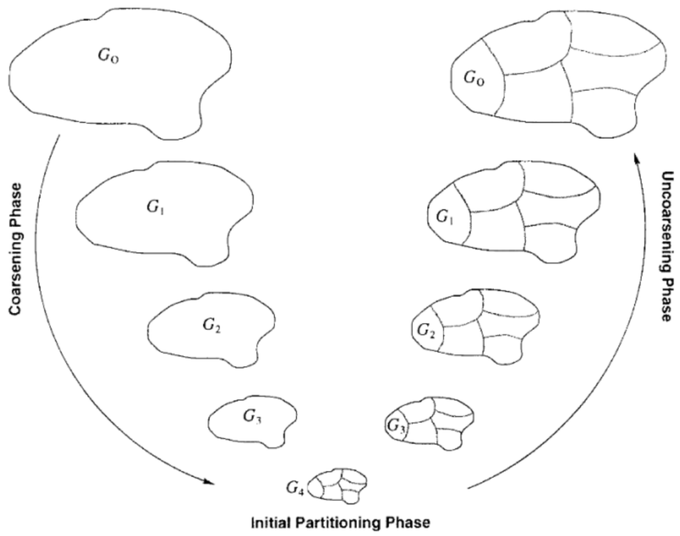
1. Przegląd literatury – SZKIELET POZIOMU 2

1.1. Metody partycjonowania grafów

1. (PODZIAŁ METOD PARTYCJONOWANIA -OD TYCH NAJMNIEJ WAŻNYCH DO NAJWAŻNIEJSZYCH) Rozbudowany podział metod został zaproponowany przez autorów artykułu - [15] oraz [22] Zgodnie z ich analizą metody dzielimy na:

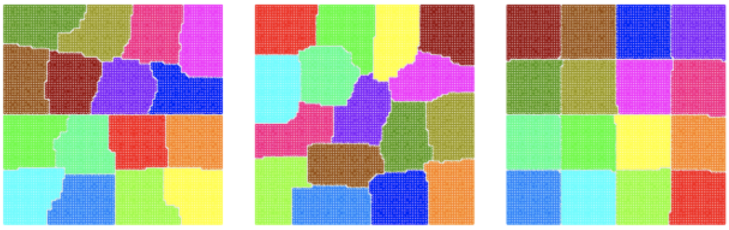
a. (TUTAJ O METODACH SPEKTRALNYCH) Metody spektralne - partycjonowanie spektralne daje dobre rezultaty i jest metodą w miarę często używaną [24, 25, 17]. Są to jednak metody kosztowne z racji na obliczanie wektorowa własnego odpowiadającego drugiej najmniejszej wartości własnej (Fielder wektor). Istnieję udane próby ulepszenia czasu wykonania tych metod, które polegają na liczeniu Fielder wektora poprzez algorytm wielopoziomowy - MSB [1]. Jednak nawet te metody wciąż charakteryzują się wysoką złożonością.
b. (TUTAJ O REKURSYWNYCH METODACH) Metody rekursywne - Są to metody często prostsze w implementacji, jednak nie sprawdzają się tak dobrze w kontekście bardziej skomplikowanych problemów głównie ze względu na to, że mają zachłanną naturę. Ponadto wykorzystanie takich algorytmów w kontekście niepodzielnych obszarów nie zdaje egzaminu, ze względu na ich specyfikę. Podejście do partycjonowania zakładające dzielenie grafu na coraz mniejsze części sprawia, że nie jesteśmy w stanie uwzględniać części niepodzielnych, lub rozwiązanie tego problemu byłoby skomplikowane. Przykład - dzielimy siatkę na 4 części. Można sobie wyobrazić sytuację, kiedy obszar niepodzielny zajmuje 50% całej siatki. Po pierwszej turze rekursywnego algorytmu mamy dwie partycje, każda zajmująca 50% powierzchni. Chcielibyśmy podzielić każdą z nich na dwie części, natomiast nie jesteśmy w stanie tego zrobić ponieważ jedna z nich jest w całości obszarem niepodzielnym. Metoda [2] zakłada żę dzielimy siatkę na liczbę obszarów, która jest równa potędze liczby dwa. Ta metoda potrafi także dzielić siatkę wedle możliwości obliczeniowych poszczególnych rdzeni procesora. Czasami metody rekursywne są implementowane jako faza metod bisekcji spektralnej [24].
c. (TUTAJ O METODACH GEOMETRYCZNYCH) Metody geometryczne - inną klasą metod są metody geometryczne [19, 27, 20, 21, 23]. Ich cechą charakterystyczną jest szybki czas wykonania, natomiast gorsze rezultaty podziału. Najlepsze wyniki spośród wyżej wymienionych metod prezentują [20, 21], natomiast z powodu losowej natury wymagane jest wielokrotne użycie algorytmu (od 5 do 50 razy) aby uzyskać wynik porównywalny z metodami spektralnymi. Wielokrotnie wywołanie zwiększa czas otrzymywania rezultatu, natomiast jest on wciąż niższy od metod spektralnych. Metody geometryczne są aplikowalne tylko w przypadku kiedy dostępne są współrzędne wszystkich wierzchołków w grafie. Dla wielu dziedzin problemów (programowanie liniowe, VLSI), nie otrzymujemy współrzędnych wraz z grafem. Istnieją algorytmy, które są w stanie obliczyć współrzędne dla wierzchołków grafu [4] wykorzystując metody spektralne ale są bardzo kosztowne i dominują czas potrzebny na samo partycjonowanie grafu.

d. (TUTAJ O METODACH WIELOPOZIOMOWYCH)
Metody wielopoziomowe [15, 28, 3, 5, 10, 9, 11, 8, 18] - cechą charakterystyczną tego podejścia jest redukcja wielkości grafu poprzez łączenie wierzchołków i krawędzi, następnie dzielenie zmniejszonego grafu na partycje, ostatnią fazą jest przywrócenie początkowego grafu zachowując podział. Często graf zmniejszany jest aż liczba wierzchołków nie osiągnie liczby partycji, którą chcemy otrzymać [22], a fazie przywracania grafu do początkowej wielkości towarzyszy algorytm, którego celem jest ulepszanie podziału [6, 7]. Algorytm ten, bazując na zmniejszonym grafie, niesie za sobą niższy koszt obliczeniowy. Jego działanie polega na zmniejszaniu długości granic pomiędzy partycjami z jednoczesnym zachowaniem ich wielkości. Na tym etapie może także zostać dodana faza balansowania, która stopniowo zmniejsza różnice w wielkości pól pomiędzy obszarami. Metody te zostały stworzone z myślą o zmniejszeniu czasu patrycjonowania kosztem jego jakości. Obecnie dają jednak bardzo dobre rezultaty również w kwestii jakości podziału. Późniejsze prace w dziedzinie tych algorytmów pokazały, że dają one lepsze rezultaty niż metody spektralne [15]. Biblioteki jak Party [22], Metis [15], Jostle [28], Chaco [11], dające state-of-the-art wyniki w kwestii jakości partycjonowania najczęściej bazują na schemacie wielopoziomowym [11].



Rysunek 1: Wielopoziomowe partycjonowanie grafu przedstawiające fazę zmniejszania grafu, następnie przypisanie partycji na zmniejszonym grafie, na końcu przywrócenie grafu to początkowej wielkości. Źródło[13].

1.2. Wybór metody



Rysunek 2: Partycjonowanie siatki 100x100 na 16 obszarów. Od lewej - pmetis[15] uzyskuje edge-cut wynoszący 688, następnie Jostle[28] z wynikiem 695 oraz Party[22] z wynikiem 615.

(TUTAJ O TYM NA JAKIE PODEJSCIE SIE ZDECYDOWAŁEM I DLACZEGO)
(WSTĘP - UZASADNIENIENIE CZEMU MUSZĘ TWORZYĆ WŁASNĄ METODE NA BAZIE GOTOWYCH METOD) Wszystkie wyżej wymienione metody nie biorą pod uwagę problemu obszarów niepodzielnych oraz obszarów wyłączonych z obliczeń.
W związku z tym moim zadaniem było znalezienie metody dającej możliwie najlepsze rezultaty w zakresie partycjonowania grafów oraz dostosowanie jej do wyżej wymienionych rozszerzeń problemu partycjonowania.
(DLACZEGO PARTY - PORÓWNANIE DO INNYCH BIBLIOTEK)

- Metody wielopoziomowe były najlepszym wyborem, z racji na to, że gwarantowały najlepsze wyniki partycjonowania
- Wiele metod wielopoziomowych do wyboru - [15, 28, 3, 5, 10, 9, 11, 8, 18], jednak brałem pod uwagę metody state-of-the-art jak Party [22], Metis [15], Jostle [28], Chaco [11], dlatego głównie na nich skupię się w tym porównaniu.
- podobieństwa i różnice - Do zmniejszenia grafu stosowane są różne warianty Matching Algorithm. Porównanie większości z nich można znaleźć tutaj [12] Przykładowo biblioteka Party [22] oraz Bui and Jones [3] stosują Maximal Weighted Matching [26]. Ze względu na wymagania co do złożoności obliczeniowej wszystkie algorytmy używają heurystyk. Wszystkie z tych metod zmniejszają graf, jednak tylko Jostle i Party zmniejsza graf, aż do otrzymania liczby wierzchołków równej liczbie partycji, na które chcemy podzielić wejściowy graf. Dzięki temu metoda partycjonowania, działająca na najmniejszym możliwym grafie jest dużo prostsza niż w pozostałych metodach. [12]

Biblioteka Party [22] okazała się dawać najlepsze rezultaty w porównaniu do innych bibliotek dających rezultaty state-of-the-art. Ponadto faza zmniejszania grafu często jest stosunkowo łatwa do zrównoleglenia [14]. Fazą, która nie podlega zrównolegleniu jest faza ulepszania istniejącego podziału. Najczęsciej bazuje ona na metodzie Fiduccia-Mattheysesa [7], która jest zoptymalizowaną pod kątem czasu działania heurystyką Kerninghan-Lin (KL) [16].

Materiały źródłowe

[1] S. Barnard and H. Simon. A fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems. pages 711–718, 01 1993.

[2] M. Berger and S. Bokhari. A partitioning strategy for nonuniform problems on multiprocessors. *IEEE Transactions on Computers*, C-36, 06 1987.

[3] T. N. Bui and C. Jones. A heuristic for reducing fill-in in sparse matrix factorization. In *PPSC*, 1993.

[4] T. F. Chan, J. R. Gilbert, and S.-H. Teng. Geometric spectral partitioning. Technical report, 1995.

[5] C.-K. Cheng and Y.-C. Wei. An improved two-way partitioning algorithm with stable performance (vlsi). *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 10(12):1502–1511, 1991.

[6] R. Diekmann and B. Monien. Using helpful sets to improve graph bisections. 08 1994.

[7] C. M. Fiduccia and R. M. Mattheyses. A linear-time heuristic for improving network partitions. In *Proceedings of the 19th Design Automation Conference*, DAC '82, page 175–181. IEEE Press, 1982.

[8] J. Garbers, H. Promel, and A. Steger. Finding clusters in vlsi circuits. In *1990 IEEE International Conference on Computer-Aided Design. Digest of Technical Papers*, pages 520–523, 1990.

[9] Hagen and Kahng. A new approach to effective circuit clustering. In *1992 IEEE/ACM International Conference on Computer-Aided Design*, pages 422–427, 1992.

[10] L. Hagen and A. Kahng. Fast spectral methods for ratio cut partitioning and clustering. In *1991 IEEE International Conference on Computer-Aided Design Digest of Technical Papers*, pages 10–13, 1991.

[11] B. Hendrickson and R. Leland. A multi-level algorithm for partitioning graphs. pages 28– 28, 02 1995.

[12] G. Karypis and V. Kumar. Analysis of multilevel graph partitioning. pages 29– 29, 02 1995.

[13] G. Karypis and V. Kumar. Multilevelk-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed Computing*, 48(1):96–129, 1998.

[14] G. Karypis and V. Kumar. A parallel algorithm for multilevel graph partitioning and sparse matrix ordering. *Journal of Parallel and Distributed Computing*, 48(1):71–95, 1998.

[15] G. Karypis and V. Kumar. Kumar, v.: A fast and high quality multilevel scheme for partitioning irregular graphs. siam journal on scientific computing 20(1), 359-392. *Siam Journal on Scientific Computing*, 20, 01 1999.

[16] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 49(2):291–307, 1970.

[17] R. Leland and B. Hendrickson. An improved spectral graph partitioning algorithm for mapping parallel computations. 16, 09 1992.

[18] N. Mansour, R. Ponnusamy, A. Choudhary, and G. C. Fox. Graph contraction for physical optimization methods: A quality-cost tradeoff for mapping data on parallel computers. In *Proceedings of the 7th International Conference on Supercomputing*, ICS '93, page 1–10, New York, NY, USA, 1993. Association for Computing Machinery.

[19] G. Miller, S. Teng, and W. Thurston. A cartesian parallel nested dissection algorithm. 1994.

[20] G. Miller, S.-H. Teng, and S. Vavasis. A unified geometric approach to graph separators. In *[1991] Proceedings 32nd Annual Symposium of Foundations of Computer Science*, pages 538–547, 1991.

[21] G. L. Miller, S. Teng, W. Thurston, and S. A. Vavasis. Automatic mesh partitioning. In A. George, J. Gilbert, and J. Liu, editors, *Graphs Theory and Sparse Matrix Computation*, The IMA Volumes in Mathematics and its Application, pages 57–84. Springer-Verlag, 1993. Vol 56.

[22] B. Monien and S. Schamberger. Graph partitioning with the party library: helpful-sets in practice. In *16th Symposium on Computer Architecture and High Performance Computing*, pages 198–205, 2004.

[23] B. Nour-Omid, A. Raefsky, and G. Lyzenga. Solving finite element equations on concurrent computers. 1987.

[24] A. Pothen, H. D. Simon, and K.-P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Anal. Appl.*, 11(3):430–452, May 1990.

[25] A. Pothen, H. D. Simon, L. Wang, and S. T. Barnard. Towards a fast implementation of spectral nested dissection. In *Proceedings of the 1992 ACM/IEEE Conference on Supercomputing*, Supercomputing '92, page 42–51, Washington, DC, USA, 1992. IEEE Computer Society Press.

[26] R. Preis. Linear time 1/2 -approximation algorithm for maximum weighted matching in general graphs. In *Proceedings of the 16th Annual Conference on Theoretical Aspects of Computer Science*, STACS'99, page 259–269, Berlin, Heidelberg, 1999. Springer-Verlag.

[27] P. Raghavan. Line and plane separators. Technical report, LAPACK WORKING NOTE 63 (UT CS-93-202), 1993.

[28] C. Walshaw and M. Cross. Mesh partitioning: A multilevel balancing and refinement algorithm. *SIAM Journal on Scientific Computing*, 22, 07 2004.