УДК 519.61 doi:10.15421/321611

О. М. Притоманова, О. С. Булавка
Дніпропетровський національний університет ім. Олеся Гончара

ЗАСОБИ ВІЗУАЛІЗАЦІЇ У ПРОГРАМНІЙ РЕАЛІЗАЦІЇ АЛГОРИТМУ ОПТИМАЛЬНОГО РОЗБИТТЯ МНОЖИН

Розроблено програмний продукт, що дозволяє розв'язувати задачу оптимального розбиття множин, основною особливістю якого є візуалізація процесу оптимального розбиття множини на підмножини із визначенням центрів підмножин.

Разработан программный продукт, который позволяет решать задачу оптимального разбиения множеств, основной отличительной чертой которого является визуализация процесса оптимального разбиения множества на подмножества с отысканием центров подмножеств.

Developed software that can solve the problem of optimal partitioning sets, the main feature of which is a graphic visualization of the process of optimal splitting the set into subsets with centers of subsets finding.

Ключові слова: задача оптимального розбиття множин, нескінченновимірна оптимізація, недиференційовна оптимізація, візуалізація.

Вступ. Неперервна задача оптимального розбиття множин актуальна з огляду на велику кількість галузей її практичного застосування. Виділяють численні методи та алгоритми розв'язання зазначеної задачі, засновані на зведенні її до задачі багатовимірної недиференційованої оптимізації та на методах недиференційовної оптимізації, зокрема на субградієнтних методах із розтягом простору у напрямку різниці двох послідовних субградієнтів (так звані r-алгоритми) [1].

Сучасні засоби й методи комп'ютерної візуалізації дозволяють проілюструвати ефективність вищезазначених алгоритмів.

Метою дослідження є розробка програмної реалізації алгоритму оптимального розбиття множин із застосуванням існуючих засобів комп'ютерної візуалізації, а також створення власного засобу візуалізації, що враховує специфіку поставленої задачі.

Для досягнення мети необхідно проаналізувати існуючі технологій комп'ютерної візуалізації, розбити їх на групи, охарактеризувати недоліки та переваги кожної з цих груп та і основі такого аналізу розробити власне програмне забезпечення для комп’ютерної візуалізації, яке має враховувати та усувати виявлені недоліки наявних технологій.

Для розробки комп’ютерної візуалізації покрокового процесу пошуку оптимального розбиття множин наведемо математичну модель для неперервної однопродуктової задачі оптимального розбиття множини на підмножини із визначенням центрів підмножин із [2].

Математична модель. Нехай Ω – обмежена вимірна за Лебегом множина в n-вимірному евклідовому просторі E_n.

Сукупність вимірних за Лебегом підмножин $\Omega_1,...,\Omega_N$ із $\Omega \subset \mathrm{E}_n$ назвемо можливими розбиттями множини Ω на її підмножини $\Omega_1,...,\Omega_N$, які не перетинаються, якщо

$$\bigcup_{i=1}^{N} \Omega_i = \Omega, \;\; mes\left(\Omega_i \cap \Omega_j\right) = 0, \; i \neq j, \, i, \, j = 1,..., N \, ,$$

де $mes(\cdot)$ означає міру Лебега.

Позначимо клас усіх можливих розбиттів множини Ω на її підмножини $\Omega_1,...,\Omega_N$, які не перетинаються, через $\sum_{\Omega}^{N}$, тобто

$$\textstyle\sum_{\Omega}^{N} = \left\{ \left(\Omega_1,...,\Omega_N\right) : \bigcup_{i=1}^{N} \Omega_i = \Omega, mes\left(\Omega_i \cap \Omega_j\right) = 0, \, i \neq j, \, i, \, j = 1,..., N \right\}.$$

Уведемо функціонал

$$F\left(\left\{\Omega_1,...,\Omega_N\right\}, \left\{\tau_1,...,\tau_N\right\}\right) = \sum_{i=1}^{N} \int_{\Omega_i} (c(x, \; \tau_i) + a_i)\rho\left(x\right)dx \, ,$$

де $x = \left(x^{(1)},...,x^{(n)}\right) \in \Omega$; $\tau_i = \left(\tau_i^{(1)},...., \tau_i^{(n)}\right) \in \Omega$; функції $c\left(x, \tau_i\right) -$ дійсні, обмежені, визначені на $\Omega \times \Omega$, вимірні за x за будь-якого фіксованого $\tau_i = \left(\tau_i^{(1)},...,\tau_i^{(n)}\right)$ із Ω для усіх $i = 1,..,N$; $\rho(x)$ $-$ обмежена, вимірна, невід’ємна на Ω функція; $a_1,...a_N$ $-$задані невід’ємні значення.

Тут та в подальшому інтеграли тлумачать за Лебегом. Вважатимемо, що міра множини граничних точок Ω_i, $i = 1,.., N$, дорівнює нулю.

Тоді під неперервною однопродуктовою задачею оптимального розбиття множини Ω на її підмножини $\Omega_1,...,\Omega_N$, які не перетинаються (серед яких можуть бути і порожні), без обмежень із визначенням координат центрів $\tau_1,..., \; \tau_N$ цих підмножин $\Omega_1,...,\Omega_N$ відповідно розумітимемо нижченаведену задачу.

Знайти

$$\min_{\{\Omega_1,\ldots,\ \Omega_N\},\{\tau_1,\ldots,\ \tau_N\}} F\left(\left\{\Omega_1,\ldots,\Omega_N\right\},\left\{\tau_1,\ldots,\tau_N\right\}\right),$$

$$\{\Omega_1,\ldots,\ \Omega_N\}\in\Sigma_\Omega^N,\quad \tau=\left(\tau_1,\ldots,\ \tau_N\right)\in\Omega^N,$$

де $x=\left(x^{(1)},\ldots,x^{(n)}\right)\in\Omega$; $\tau_i=\left(\tau_i^{(1)},\ldots,\tau_i^{(n)}\right)\in\Omega$.

Перепишемо задачу в термінах характеристичних функцій $\lambda_i(x)$ підмножин Ω_i , $i=1,\ldots,\ N$, у такому вигляді: обчислити

$$\min_{(\lambda(\cdot),\tau)\in\Gamma_1\times\Omega^N}\int_\Omega\sum_{i=1}^N(c(x,\ \tau_i)+a_i)\rho\left(x\right)\lambda_i\left(x\right)dx,$$

де

$$\Gamma_1=\left\{\lambda(x)=(\lambda_1(x),\ldots,\lambda_N(x))\right.$$

$$\left.\sum_{i=1}^N\lambda_i(x)=1\ \text{майже всюди (м.в.) для}\ x\in\Omega\right\};$$

$$\lambda_i(x)=0\vee 1\ \text{м.в. для}\ x\in\Omega,\ i=1,\ldots,N,$$

$$\tau=\left(\tau_1,\ldots,\tau_N\right)\in\underbrace{\Omega\times\ldots\times\Omega}_{N}=\Omega^N.$$

Існуючі технології візуалізації. Для розробки комп’ютерної візуалізації покрокового процесу пошуку оптимального розбиття множин важливий правильний вибір засобів і технологій з огляду на доступність, зручність та ефективність їх застосування. Розглянемо різні підходи до розв’язання поставленої задачі, упорядкувавши їх за зростанням складності реалізації.

Візуалізація об’єктів засобами прикладних математичних пакетів. Найбільш очевидним і поширеним засобом є застосування сторонніх розробок. На ринку програмного забезпечення беззаперечними лідерами є декілька математичних пакетів: Mathematica, Mapple, MathCad, MatLab, а також безліч невеликих пакетів, спрямованих на розв’язання вузького кола задач. Ці засоби надають широкі можливості для візуалізації математичних об’єктів і понять. Очевидними перевагами даного підходу є загальнодоступність, різноманіття і в загальному випадку гарантована стабільність роботи програмних продуктів. Проте різноманіття програмного забезпечення потребує значних ресурсів для вивчення та пошуку найбільш придатного пакета. Гарантована стабільність часто потребує значних витрат. До того ж, не можна бути впевненим у тому, що в конкретному математичному пакеті буде саме така візуалізація, яка задовольнить потреби викладача.

Візуалізація об'єктів засобами інтерактивної комп'ютерної графіки. Анімація як засіб ілюстрації понять стала загальнодоступною з розвитком комп'ютерних і мультимедіа-технологій. Перехід від по кадрового прорисовування до опису поведінки векторних об'єктів значно скоротив працю аніматорів, але наклав низку обмежень на зображувальні засоби. Математика, яка оперує точними графічними побудовами, від такого переходу тільки виграла.

Найпопулярнішим засобом комп'ютерної анімації безперечно є технологія векторної графіки Flash [3]. Гнучкі засоби середовища розробки flash-анімації дозволяють автоматизовувати зображення анімованого переміщення та змінювати вигляд об'єктів, а також умонтовувати в ролики елементи інтерактивності. Безсумнівною перевагою застосування Flash також є можливість повної сумісності flash-роликів із мовами гіпертекстової розмітки. Ілюстрація поняття за допомогою проектора перед аудиторією може бути здійснена будь-якими засобами анімації, але якщо мова йде про створення електронних підручників, що містять ілюстрацію понять, то пріоритет надають web-технологіям, однією із яких є Flash. Проте технологія Flash має доволі очевидні недоліки. Якщо ідеться про ілюстрування простих математичних понять (графіки функцій, діаграми Ейлера, проміжки знакопостійності тощо), то Flash дуже ефективно виконує ці завдання. Якщо ж є потреба у візуалізації тривимірного об'єкта (а саме такі важкі побудови часто потрібні викладачу від), то застосування Flash вкрай неефективне: розробник або буде приблизно будувати ізометрію простору, або ж займатиметься розрахунком проекцій. Такі підходи виправдані в разі побудови статичного рисунка, але не у випадках, коли ідеться про анімацію об'єктів або динамічну зміну вигляду.

Для анімації тривимірних візуалізацій можна також застосовувати складніші прикладні графічні пакети (наприклад, 3DsMax), але у випадку застосування цього підходу втрачається інтерактивність візуалізації [4].

Візуалізація об'єктів засобами інструментальних графічних бібліотек. Метод побудови дво- і тривимірних об'єктів засобами програмних інтерфейсів (API) комп'ютерної графіки значно складніший за вищезазначені, але за вмілого застосування можна отримати відмінні результати [5].

Серед таких засобів виділяють DirectX та OpenGL, причому перевагу частіше надають останньому через відкритий код і багатоплатформеність. OpenGL надає розробнику набір базових графічних примітивів для дво- і тривимірних побудов, а середовище розробки дозволяє залучити потужні обчислювальні засоби для математичних розрахунків [6].

Засобами OpenGL дуже наочно (із елементами динаміки й інтерактивності) ілюструють такі складні математичні поняття, як криві та поверхні у просторі, визначений інтеграл й інтеграли від функцій багатьох змінних, граничну поведінку функцій.

Основні результати роботи розробленого програмного забезпечення. Після вивчення усіх вищезазначених способів візуалізації було зроблено висновок, що поставлене завдання щодо розбиття множин унікальне для існуючих засобів комп'ютерної візуалізації. Саме тому було ухвалено рішення розробити власне програмне забезпечення, яке дозволить налаштовувати параметри алгоритму та наочно демонструвати покроковий процес його виконання.

Як платформу для розробки обрано браузер через його доступність і поширеність. Таким чином, усі вхідні параметри алгоритму задають на HTML формі. Як інструмент візуалізації поточного стану розбиття спочатку було обрано canvas, але синтетичні тести довели, що генерація bitmap зображення та його передача об'єкту img має на 10 % більшу швидкість. Такий ефект можна пояснити тим, що зображення складається з великих ділянок одного кольору – підмножин, на які розбивається вхідна множина.

Отже, алгоритм було запрограмовано на мові JavaScript останнього стабільного стандарту ES6 (2015). Для роботи з матрицями використано бібліотеку MathJS.

На рис. 1–3 наведено результати роботи розробленого програмного забезпечення для візуалізації покрокового процесу пошуку оптимального розбиття для неперервної однопродуктової задачі із визначенням центрів підмножин.

На рис. 1 зображено 18 кадрів анімації покрокового процесу оптимального розбиття множини на 3 підмножини із заданими центрами підмножин, зміну яких користувач програмного забезпечення побачить досить видко. У разі зменшення крокового множника алгоритму сусідні зображення стануть більш схожими між собою. За рахунок цього можна досягти плавності анімації, тому покрокове перетворення підмножин і їх центрів у динаміці користувач сприйматиме природно.

На рис. 2 зображено 24 кадри анімації покрокового процесу оптимального розбиття множини на 4 підмножини, а на рис. 3 – 30 кадрів для візуалізації оптимального розбиття множини на 7 підмножин.

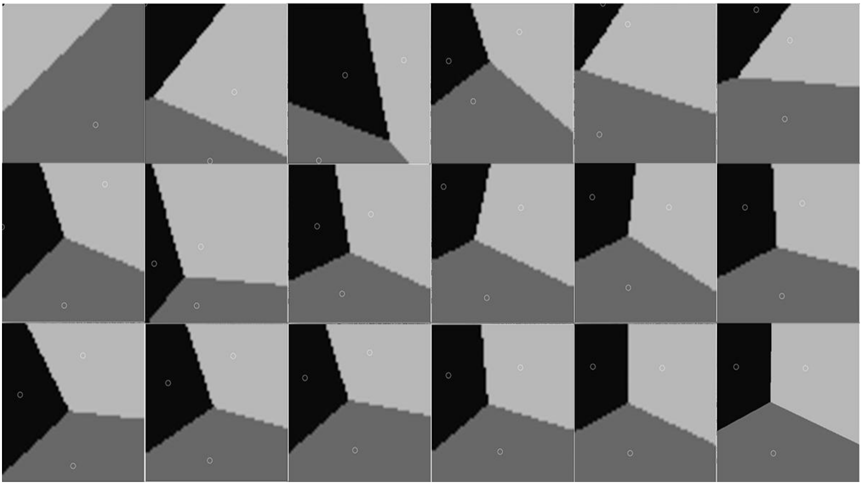

Рис. 1. **Послідовність кадрів анімації покрокового процесу оптимального розбиття множини на 3 підмножини із заданими центрами підмножин**

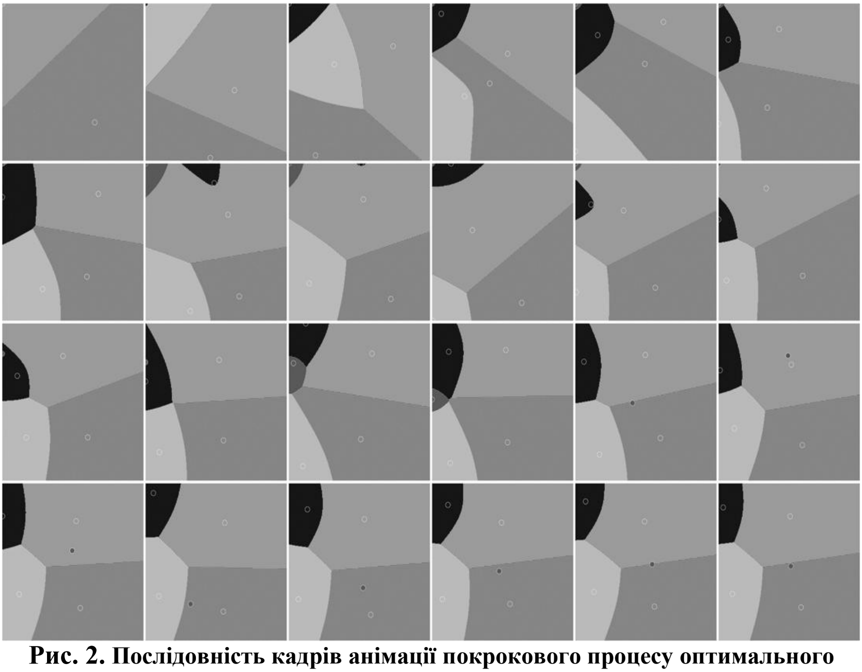

Рис. 2. **Послідовність кадрів анімації покрокового процесу оптимального розбиття множини на 4 підмножини із заданими центрами підмножин**

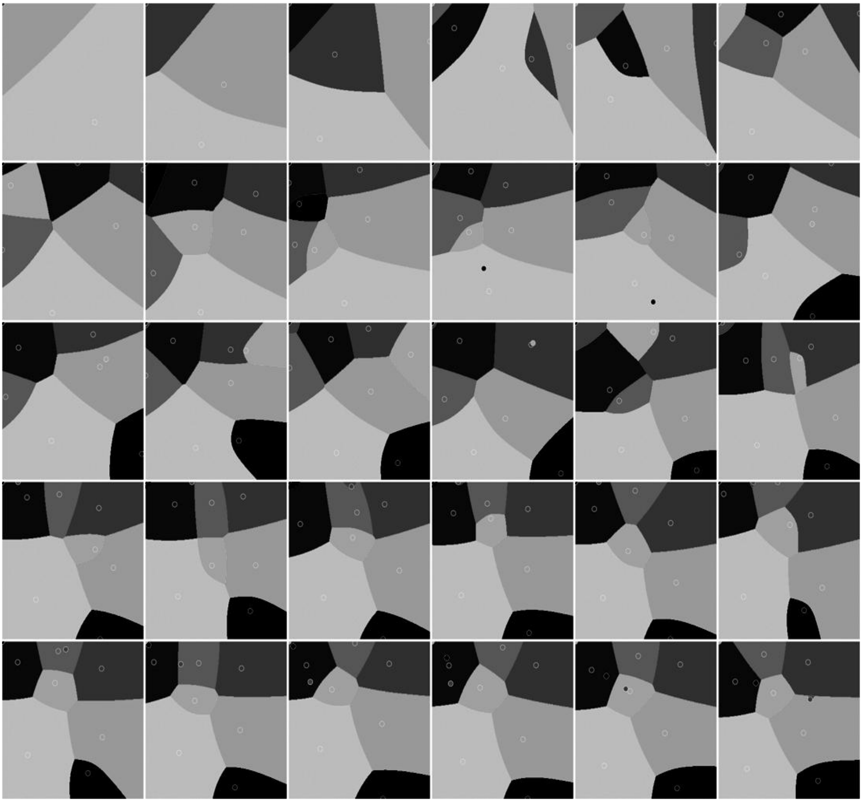

Рис. 3. Послідовність кадрів анімації покрокового процесу оптимального розбиття множини на 7 підмножини із заданими центрами підмножин

Висновки. Розроблене програмне забезпечення для візуалізації роботи алгоритмів покрокового процесу пошуку оптимального розбиття множини на підмножини може бути використане:

1. Для навчальних цілей. Вдала візуалізація процесу сприятиме швидшому засвоєнню ідей алгоритму.
2. Під час розробки алгоритму. Помилки в ході програмування складного алгоритму неминучі. Замість перевірки коректності результатів у текстовому вигляді можна першочергово розробити візуалізацію та перевіряти коректність програмної реалізації за її допомогою.
3. Для удосконалення алгоритму. Візуалізація процесу роботи може допомогти виявити неефективні аспекти алгоритму.

Бібліографічні посилання

1. **Киселева, Е.М.** Непрерывные задачи оптимального разбиения множеств и r-алгоритмы [Текст]: монография / Е.М. Киселева, Л.С. Коряшкина. – К.: Наук. думка, 2015. – 400 с.
2. **Киселева, Е.М.** Непрерывные задачи оптимального разбиения множеств [Текст]: монография / Е.М. Киселева, Н.З. Шор. – К.: Наук. думка, 2005. – 564 с.
3. **Rauschmayer A.** Exploring ES6: Upgrade to the next version of JavaScript [Text] / A. Rauschmayer – 2016. – 620 c.
4. **Sellers, G.** OpenGL SuperBible: Comprehensive Tutorial and Reference [Text]/ G. Sellers, R. S. Wright, Jr. N. Haemel – Addison-Wesley Professional – 2013. – 848 c.
5. Visualising data structures and algorithms through animation [Електронний ресурс]. – Режим доступу: http://visualgo.net. – Заголовок з екрана.
6. **Bostock, M.** Visualizing Algorithms [Електронний ресурс] / M. Bostock. – Режим доступу: https://bostocks.org/mike/algorithms/. – Заголовок з екрана.

Надійшла до редколегії 16.09.2016