ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

Hà Thị Kim Dung

MỘT SỐ THUẬT TOÁN XẤP XỈ CHO BÀI TOÁN TỐI ƯU HÀM DẠNG SUBMODULAR VỚI RÀNG BUỘC

Ngành: Khọc học máy tính
Mã số: 9480101

TÓM TẮT LUẬN ÁN TIẾN SĨ KHOA HỌC MÁY TÍNH

Hà Nội – 2024

Công trình được hoàn thành tại:

Đại học Công nghệ, Đại học Quốc gia Hà Nội

Người hướng dẫn khoa học:

1. PGS. TS Hoàng Xuân Huấn
2. TS Phạm Văn Cảnh

Phản biện: PGS. TS Huỳnh Thị Thanh Bình, Đại học Bách Khoa Hà Nội

Phản biện: PGS. TS Hà Minh Hoàng, trường Đại học Kinh tế quốc dân

Phản biện: TS Lê Xuân Thanh, Viện Toán học, Viện Hàn lâm Khoa học và Công nghệ Việt Nam

Luận án sẽ được bảo vệ trước Hội đồng cấp Đại học Quốc gia chấm luận án tiến sĩ họp tại .....

vào hồi giờ ngày tháng năm

Có thể tìm hiểu luận án tại:

- Thư viện Quốc gia Việt Nam
- Trung tâm Thông tin - Thư viện, Đại học Quốc gia Hà Nội

Hà Nội – 2024

MỤC LỤC

DANH MỤC CÁC TỪ VIẾT TẮT

Từ viết tắt	Tiếng Anh	Tiếng Việt
CO	Combinatorial Optimizationn	Tối ưu tổ hợp
DRSC	DR Submodular Cover	Tập phủ trên lưới nguyên
DS	Deterministic Streaming	Luồng tất định
IM	Influence Maximization	Tối đa ảnh hưởng
IC	Independent Cascade	Bậc độc lập
kCMK	k-topic information Coverage Maximization under Knapsack constraint	Tối đa độ phủ thông tin k-chủ đề với ràng buộc chi phí
kIMK	k-topic Influence Maximization under Knapsack constraint	Tối đa ảnh hưởng k-chủ đề với ràng buộc chi phí
kSMK	k-submodular Maximization under Knapsack constraint	Tối đa hàm k-Submodular với ràng buộc chi phí
kSPK	k-Sensor Placement under Knapsack constraint	Đặt k sensor cảm biến với ràng buộc chi phí
LT	Linear Threshold	Ngưỡng tuyến tính
MXH	-	Mạng xã hội
NCS	-	Nghiên cứu sinh
RIS	Revese Influence Sampling	Lấy mẫu ảnh hưởng ngược
RR	Reachable Reverse	Ảnh hưởng ngược
RS	Random Streaming	Luồng ngẫu nhiên
s.t	Subject to	Sao cho
SC	Submodular Cover	Tập phủ Submodular
SM	Submodular Maximization	Tối đa hàm Submodular
SMK	Submodular Maximization under Knapsack constraint	Tối đa hàm Submodular với ràng buộc chi phí
SMKN	Submodular Maximization subject to a Knapsack constraint under Noises	Tối đa hàm Submodular với ràng buộc chi phí có nhiễu

MỞ ĐẦU

Tối ưu tổ hợp là một công cụ cơ bản được ứng dụng trong nhiều lĩnh vực khoa học, kỹ thuật, y học, kinh tế..., đặc biệt là khoa học máy tính.

Trong các bài toán tối ưu tổ hợp, có nhiều bài toán có hàm mục tiêu là một dạng hàm thu thập và xử lý thông tin. Khi đó, yêu cầu đặt ra là cần phải thu thập được càng nhiều thông tin phong phú, đa dạng càng tốt. Những bài toán như vậy thường dẫn đến tìm lời giải cho *bài toán tối ưu tổ hợp với hàm mục tiêu dạng submodular*, chẳng hạn, các bài toán tóm tắt tài liệu tự động, bài toán trích chọn đặc trưng, phân tích và tiền xử lý dữ liệu, tối đa ảnh hưởng trên mạng xã hội, đặt các cảm biến... Do vậy, chủ đề nghiên cứu về tối đa hàm submodular và các biến thể của nó là một chủ đề nóng, thu hút rất nhiều các nhà khoa học quan tâm nghiên cứu và công bố tại các Hội nghị hàng đầu về trí tuệ nhân tạo và học máy như IJCAI, AAAI, NEURIPS, ICML, SODA, STOC... cũng như tại các tạp chí nổi tiếng về học thuật về tối ưu hoá, vận trù học và tối ưu tổ hợp...

Bài toán tối đa hàm submodular yêu cầu tìm lời giải $S \subseteq V$ sao cho hàm $f(\cdot)$ submodular đã cho đạt giá trị cực đại. Trong đó, việc xem xét tìm lời giải sao cho tối đa hàm mục tiêu trong điều kiện có ràng buộc được quan tâm hơn cả. Lý do vì các ràng buộc được đưa vào khiến cho bài toán gần gũi với các yêu cầu thực tế như nguồn nhân lực, tiền của, thời gian... luôn bị hạn chế. Từ đó, hình thành một lớp các bài toán như tối đa hàm submodular với ràng buộc lực lượng, với ràng buộc matroid, hoặc với ràng buộc chi phí...

Hiện nay, giải các bài toán tối ưu hàm submodular cần các thuật toán nhanh, hiệu quả trở nên vô cùng cấp thiết. Vì dữ liệu đầu vào cần xử lý ngày càng tăng nhanh, làm cho không gian tìm kiếm lời giải trở nên khổng lồ. Do vậy, việc tìm lời giải chính xác trở nên bất khả thi. Vì thế, *xu hướng đề xuất các thuật toán xấp xỉ cho lời giải cạnh tranh* với các đảm bảo lý thuyết về độ phức tạp thời gian, độ phức tạp bộ nhớ... đang chiếm ưu thế. Thuật toán xấp xỉ cho thấy ưu thế khi chỉ ra được tỉ lệ xấp xỉ của lời giải so với lời giải tối ưu là bao nhiêu, đồng thời phân tích so sánh được các đảm bảo lý thuyết khác giữa các công bố có liên quan.

Một lý do quan trọng nữa khi đề xuất các thuật toán xấp xỉ giải quyết bài toán tối đa hàm submodular là cách tiếp cận này có tính tổng quát. Các thuật toán đề xuất có thể áp dụng trên nhiều bộ dữ liệu khác nhau. Bài toán đã có mô hình tính toán, thuật toán xây dựng từ nó, các phép suy dẫn... đều đã được chứng minh tính chặt chẽ, đúng đắn bằng các bổ đề, định lý và hệ quả.

Xuất phát từ bối cảnh phát triển, cùng với các nhu cầu về phân tích dữ liệu của kỷ nguyên số, NCS và cộng sự đã tập trung nghiên cứu đề tài: "***Một số thuật toán xấp xỉ cho bài toán tối ưu hàm dạng submodular với ràng buộc***". Từ các vấn đề nghiên cứu trên đây, có thể thấy rằng có nhiều bài toán tối ưu hàm submodular được quan tâm nghiên cứu hiện nay. Trong đó, có ba bài toán tối ưu hàm submodular có ràng buộc quan trọng và có giá trị ứng dụng trong thực tiễn, còn gọi là các bài toán biến thể của tối đa hàm submodular, được luận án tập trung nghiên cứu:

1. ***Bài toán tối đa hàm k-submodular với ràng buộc chi phí*** (k-Submodular Maximization under Knapsack constraint - kSMK). Sự chuyển hướng đa dạng nghiên cứu với k-submodular được quan tâm vì phù hợp với nhiều lớp bài toán mà hàm submodular chưa đủ để giải quyết, như tập các phần tử được phân loại thành các tập con khác nhau, hoặc được chọn từ nhiều nguồn tài nguyên khác nhau... Các nhà nghiên cứu đã mở rộng hàm submodular thành k-submodular ($k \geq 2$). Khi đó, hàm mục tiêu sẽ được mở rộng thành $f : (k+1)^V \mapsto \mathbb{R}_+$. Các ứng dụng của bài toán tối đa hàm k-submodular có thể áp dụng vào tối đa ảnh hưởng của k chủ đề, tối đa thông tin lan truyền của k chủ đề, đặt k loại cảm biến... Được mở rộng từ bài

toán *tối đa hàm submodular với ràng buộc chi phí (SMK)*, bài toán kSMK xem xét tối đa hàm mục tiêu k-submodular với ràng buộc chi phí cho trước.

2. ***Bài toán tối đa hàm submodular với ràng buộc chi phí có nhiễu*** (Submodular Maximization subject to Knapsack constraint under Noises -SMKN). Bài toán tối đa hàm submodular với ràng buộc chi phí, SMK, được xem là tổng quát hơn so với ràng buộc lực lượng. Ràng buộc này yêu cầu mỗi một phần tử e trong tập dữ liệu đầu vào V sẽ có một chi phí dương $c(e) > 0$ để hoạt động, do vậy, cần phải tìm lời giải sao cho giá trị lợi ích $f(\cdot)$ thu được là lớn nhất mà tổng chi phí của tập lời giải $c(S)$ không vượt quá ngân sách cho trước. Nghiên cứu giải bài toán SMK có khả năng áp dụng vào nhiều trường hợp trong thực tế khi kinh phí, thời gian hay con người bị giới hạn. Tuy nhiên, việc tính toán chính xác hàm mục tiêu f là bất khả thi, nên cần ước lượng xấp xỉ F với sai số $\epsilon \in (0, 1)$ cho nó. F còn gọi là ước lượng nhiễu của f. Tuy nhiên, các nghiên cứu hiện nay chưa xử lý nhiễu khi tính toán với F. Cho nên luận án đặt vấn đề giải bài toán SMK nhưng có xử lý nhiễu qua bài toán SMKN.

3. ***Bài toán Phủ Submodular trên lưới nguyên*** (DR-Submodular Cover over integer lattice - DRSC). Trong khoa học máy tính, bài toán đối ngẫu của tối đa hàm submodular là Phủ Submodular (Submodular Cover - SC) cũng có nhiều ứng dụng có giá trị. Nếu như tối đa hàm submodular tương đương với tìm phủ cực đại, thì SC là bài toán tìm tập phủ tối thiểu sao cho giá trị hàm $f(\cdot)$ không thấp hơn một ngưỡng $\alpha > 0$. Giá trị của bài toán SC thể hiện khi nó tổng quát hóa các trường hợp không đòi hỏi lợi ích thu về cao nhất mà chỉ cần đạt mức nào đó, nhưng phải đảm bảo tối ưu về chi phí con người, ngân sách, thời gian... Tuy nhiên, SC xét hàm f là hàm tập hợp chưa giải quyết được tình huống *một phần tử tốt có thể được chọn đi chọn lại nhiều lần*. Ví dụ đại lý bán chạy thì nhà phân phối sẽ chọn đi chọn lại để phân phối sản phẩm. Các trường hợp tương tự như vậy cần đưa bài toán lên lưới nguyên (integer lattice) để giải. Hàm submodular mở rộng trên lưới nguyên với $f : 2^V \mapsto \mathbb{R}_+$ trở thành $f : \mathbb{Z}^V \mapsto \mathbb{R}_+$. Submodular đã được mở rộng thành *DR-submodular* và *lattice submodular* trên lưới nguyên với một số khác biệt về tính chất. Bài toán SC lúc này được tổng quát thành *Phủ DR-submodular (DR-Submodular Cover - DRSC)*.

Theo xu thế, luận án lựa chọn hướng nghiên cứu các thuật toán xấp xỉ giải bài toán tối đa hàm submodular có ràng buộc. Tuy nhiên, thách thức chung của hướng nghiên cứu này là:

- Các thuật toán phải phải đối mặt với vấn đề thời gian chạy khi dữ liệu đầu vào tăng lên.
- Các bài toán tối đa hàm submodular thường là các bài toán NP-khó, NP-đầy đủ, thậm chí việc tính toán hàm mục tiêu còn là #P-Khó (bài toán tối đa ảnh hưởng). Các bài toán này đều khó tìm lời giải trong thời gian đa thức.
- Nhiều ứng dụng trong thực tiễn cần mở rộng hoặc biến thể bài toán tối ưu hàm submodular và cần các phương pháp giải phù hợp.

Đặc biệt, khi lựa chọn ba bài toán nghiên cứu kSMK, SMKN và DRSC, mỗi bài toán lại có những thách thức riêng:

1. ***Đối với bài toán kSMK***: Sự khác nhau về bản chất của hàm submodular và hàm k-submodular dẫn tới áp dụng phương pháp đã có với submodular sang k-submodular chưa chắc đã khả thi hoặc hiệu quả bị giảm xuống; Ràng buộc chi phí làm cho có nhiều lời giải dự tuyển với nhiều chi phí khác nhau, việc chọn giải pháp tốt nhất giữa rất nhiều giải pháp làm ảnh hưởng đến thời gian chạy của thuật toán; Vấn đề giảm độ phức tạp truy vấn, góp phần giải quyết vấn đề thời gian chạy cần nhiều đóng góp mới; ấn đề hàm mục tiêu không đơn điệu trở nên thách thức hơn do phải nhanh chóng loại bỏ các phần tử làm giảm chất lượng hàm mục tiêu.

2. ***Đối với bài toán SMKN***: Bài toán SMKN là một bài toán mới, chưa có công bố nào về bài toán này; Thách thức của ràng buộc chi phí (giống bài toán kSMK) và ước lượng nhiễu F của hàm mục tiêu f; Vấn đề thời gian chạy và không gian lưu trữ.

3. ***Đối với bài toán DRSC***: Hàm submodular trên lưới nguyên có sự biến đổi về tính chất. Đây là một vấn đề còn rất mới, các nghiên cứu về nó chưa nhiều; DRSC yêu cầu tìm kiếm lời giải trong không gian $\mathbb{Z}$ chiều, lớn hơn rất nhiều so với không gian 2 chiều của hàm tập hợp submodular. Do vậy, việc đề xuất thuật toán xấp xỉ cạnh tranh với độ phức tạp truy vấn tuyến tính rất khó triển khai trên lưới nguyên.

Như vậy, ba bài toán nêu trên đều là các bài toán có giá trị nghiên cứu và còn tồn tại nhiều thách thức. Vì thế, luận án đã chọn chủ đề nghiên cứu: "***Một số thuật toán xấp xỉ cho bài toán tối ưu hàm dạng submodular với ràng buộc***". Mục tiêu cụ thể là:

1. Nghiên cứu bài toán 1 - kSMK với các mục tiêu: Đề xuất các thuật toán xấp xỉ giải quyết bài toán kSMK; Giải bài toán tổng quát. Các thuật toán đề xuất cho lời giải với tỉ lệ xấp xỉ hằng số cạnh tranh, và giảm độ phức tạp truy vấn xuống còn tuyến tính hoặc giả tuyến tính; Thực nghiệm trên một vài kịch bản cho thấy sự tiệm cận giữa lý thuyết với thực nghiệm.

2. Nghiên cứu bài toán 2 - SMKN với các mục tiêu: Xây dựng mô hình bài toán tối đa hàm submodular với ràng buộc chi phí dưới sự tác động của 2 loại nhiễu: nhiễu cộng và nhiễu nhân; Đề xuất các thuật toán Luồng xấp xỉ hiệu quả giải quyết bài toán SMKN. Thuật toán cải tiến cần cho tỉ lệ xấp xỉ cạnh tranh nhưng giảm thời gian chạy và dung lượng lưu trữ các phần tử so với thuật toán tham lam; Ước lượng xấp xỉ hàm mục tiêu trong môi trường nhiễu; Thực nghiệm trên một vài kịch bản cho thấy sự tiệm cận giữa lý thuyết với thực nghiệm.

3. Nghiên cứu bài toán 3 - DRSC với các mục tiêu: Nghiên cứu tính chất DR-submodular và mô hình hóa bài toán DRSC; Đề xuất thuật toán xấp xỉ hiệu quả giải quyết bài toán DRSC. Với bài toán này, hướng thuật toán đề xuất dựa trên thiết kế song song với độ phức tạp truy vấn và độ phức tạp song song thấp.

Luận án đã thể hiện những đóng góp sau:

1. Đề xuất các thuật toán xấp xỉ giải quyết bài toán kSMK trong hai trường hợp hàm mục tiêu k-submodular đơn điệu và không đơn điệu. Với hàm đơn điệu, luận án đề xuất 03 thuật toán xấp xỉ với độ phức tạp truy vấn tuyến tính. Trong đó, thuật toán tốt nhất nâng được tỉ lệ xấp xỉ so với thuật toán tốt nhất hiện nay mà vẫn giảm độ phức tạp truy vấn xuống một hệ số. Với hàm mục tiêu không đơn điệu, luận án đưa ra được một thuật toán cải tiến cho tỉ lệ xấp xỉ tốt tương đương thuật toán xấp xỉ tốt nhất hiện nay nhưng giảm độ phức tạp truy vấn xuống một hệ số.

Các thực nghiệm cho thấy các thuật toán của luận án cho chất lượng lời giải cạnh tranh, giảm thời gian chạy đáng kể, đặc biệt khi dữ liệu đầu vào và ngân sách tăng lên. Các kết quả nghiên cứu đã được công bố tại: hội thảo quốc tế *the 9th IEEE International Conference on Data Science and Advanced Analytics* (DSAA), 2022 (**RANK A**), tạp chí *Computers & Operations Research* (**ISI/Q1**), và hội thảo quốc tế *the 15th IEEE International Conference on Knowledge and Systems Engineering (KSE)*, 2023.

2. Đề xuất các thuật toán xấp xỉ giải quyết bài toán SMKN với hàm mục tiêu submodular đơn điệu. Luận án đề xuất 02 thuật toán: (1) thuật toán tham lam; (2) thuật toán luồng cải tiến thuật toán tham lam với các đảm bảo lý thuyết tương đương và giải quyết vấn đề thời gian chạy, không gian lưu trữ. Với nghiên cứu này, luận án cũng trình bày phần thực nghiệm cho thấy thuật toán cải tiến cho giá trị hàm mục tiêu tốt gần bằng tham lam, trong khi thời gian chạy và bộ nhớ tiết kiệm hơn. Các kết quả nghiên cứu đã được công bố ở tạp chí *Asia-Pacific Journal of Operational Research*, tập 39, số 6, 2022 (**ISI/Q3**).

3. Đề xuất một thuật toán xấp xỉ tiêu chí kép được thiết kế song song hoá giải quyết hai lớp bài toán, DRSC và SC. Thuật toán cho chất lượng lời giải tốt tương đương với thuật toán tiêu chí kép tốt nhất hiện nay cho DRSC, song giảm được đáng kể số lượng truy vấn, số lượng vòng tuần tự, qua đó giảm được thời gian chạy. Các kết quả nghiên cứu đã được công bố tại tạp chí *Information Processing Letters*, tập 182, 2023 (**ISI/Q3**).

Ngoài phần mở đầu và kết luận, luận án chia thành 04 chương:

Chương 1 trình bày các nghiên cứu khái quát về bài toán tối ưu hàm dạng submodular. Chương 2 trình bày các kết quả nghiên cứu đối với bài toán tối đa hàm k-submodular với ràng buộc chi phí (Bài toán kSMK). Chương 3 trình bày các kết quả nghiên cứu đối với bài toán tối đa hàm submodular với ràng buộc chi phí trong môi trường có nhiễu (Bài toán SMKN). Chương 4 trình bày các kết quả nghiên cứu đối với bài toán Phủ Submodular trên lưới nguyên (DRSC).

CHƯƠNG 1
BÀI TOÁN TỐI ƯU TỔ HỢP HÀM DẠNG SUBMODULAR

Lớp bài toán quan trọng trong khoa học máy tính là *Tối ưu tổ hợp (Combinatorial Optimization - CO)*. Tuy nhiên, các bài toán CO hầu hết là NP-khó, NP-đầy đủ nên các kỹ thuật giải chính xác khó tìm được lời giải trong thời gian đa thức, cần phải tìm lời giải gần đúng, như các thuật toán dựa trên kinh nghiệm (heuristic và metaheuristic) và các thuật toán xấp xỉ. Nhược điểm của các phương pháp tìm kiếm dựa trên kinh nghiệm là không chỉ ra đảm bảo lý thuyết so với lời giải tốt nhất thì lời giải này đúng được bao nhiêu %. Đôi khi, thiết kế một thuật toán có thể dẫn tới kết quả tồi, ảnh hưởng tới thời gian kiểm nghiệm, tinh chỉnh các tham số của thuật toán. *Thuật toán xấp xỉ (approximation algorithm)* với các đảm bảo lý thuyết được chỉ rõ khắc phục được các nhược điểm nói trên.

Luận án nghiên cứu giải bài toán CO hàm dạng submodular bằng các thuật toán xấp xỉ. Hàm submodular mới được nghiên cứu trong vòng 20 năm và trở nên thu hút trong thời gian gần đây do tính chất độc đáo của hàm mục tiêu submodular làm cho nó có thể được ứng dụng vào nhiều bài toán khác nhau trong thực tiễn. Trong đó, bài toán *tối đa hàm submodular,* SM, được đánh giá là một trong các bài toán đặc biệt nhất khi có thể nhanh chóng thu thập và xử lý thông tin mà không tạo ra sự dư thừa, lãng phí hay làm mất mát thông tin.

1.1. Bài toán CO tối đa hàm submodular

1.1.1. Phát biểu bài toán

Bài toán SM phát biểu dưới dạng một bài toán CO như sau:

Định nghĩa 1.1 (Bài toán SM)**.** Cho tập cơ sở V và hàm $f : 2^V \mapsto \mathbb{R}_+$ là hàm tập hợp submodular (với $f(\emptyset) = 0$), bài toán cần tìm:

$$\max f(S)$$
$$\text{s.t } S \subseteq V,$$

với $\mathcal{C}$ là ràng buộc cho trước của bài toán.

Một số trường hợp phổ biến của $\mathcal{C}$ là:

- $\mathcal{C} = \emptyset$: bài toán không ràng buộc;

- $\mathcal{C} = \{|S| \leq k\}$: ràng buộc lực lượng;
- $\mathcal{C} = \{\forall A \subseteq B \subseteq V, f(A) \leq f(B)\}$: polymatroid (hàm f có tính đơn điệu)};
- $\mathcal{C} = \{c(S) \leq B\}$ với $c(S)$ là chi phí của tập S, B là ngân sách cho trước, đây là ràng buộc chi phí.

Một lưu ý khi giải các bài toán SM, đó là người ta luôn giả sử đã cho trước cách tính hàm f. Nói cách khác, tồn tại một hộp đen sao cho với mọi đầu vào $S \subseteq V$ qua hộp đen này đều tính được $f(S)$. Giá trị $f(S)$ lúc này được gọi là giá trị *ước lượng (oracle)* của hàm mục tiêu, hoặc nói ngắn gọn là giá trị của hàm mục tiêu. Hiện nay, người ta rất quan tâm đến số lời gọi các ước lượng này và đưa ra một phép đo quan trọng để đánh giá độ phức tạp của thuật toán là *số lượng truy vấn (query)* được thể hiện bằng *độ phức tạp truy vấn (query complexity)*.

Điều làm nên sự đặc biệt của bài toán SM là tính chất submodular của hàm mục tiêu. Phần tiếp theo sẽ nghiên cứu sâu hơn về hàm này.

1.1.2. Hàm mục tiêu submodular

Một số phát biểu tính chất thường dùng của hàm submodular:

Định nghĩa 1.2 (Hàm submodular)**.** Cho V là một tập hữu hạn không rỗng được gọi là tập cơ sở, hàm $f : 2^V \mapsto \mathbb{R}_+$ là một hàm submodular khi và chỉ khi nó thỏa mãn bất đẳng thức sau:

$$\forall X, Y \subseteq V : f(X) + f(Y) \geq f(X \cap Y) + f(X \cup Y).$$

Không giảm tổng quát, hàm $f : 2^V \mapsto \mathbb{R}_+$, tức là $f(S) \geq 0, \forall S \subseteq V$. Thêm vào đó, luôn có giả sử $f(\emptyset) = 0$ biểu thị đối với tập rỗng, hàm không thu được một lợi ích nào.

Khi nghiên cứu, hàm submodular được chỉ ra có sự phù hợp với *tính chất lợi nhuận hiệu suất giảm dần*, một tính chất rất nổi tiếng thường dùng trong kinh tế học. Giả sử tập $A \subseteq V$ là một tập các người dùng hoặc hoạt động đầu tư nào đó cho lợi ích là $f(A)$. Hàm submodular f mang ý nghĩa sau khi thực hiện một loạt các hành động của tập A, *lợi nhuận biên (marginal gain)* của bất kì một phần tử e nào đó sẽ không tăng khi thực hiện các hoạt động trong tập $V \setminus A$. Do đó, hàm submodular trên tập hợp là hàm có tính chất *lợi nhuận hiệu suất giảm dần (diminishing returns property)* tự nhiên được phát biểu như sau:

Định nghĩa 1.3. (Tính chất lợi nhuận hiệu suất giảm dần của hàm submodular) Hàm tập hợp $f : 2^V \mapsto \mathbb{R}_+$ là hàm submodular và các tập $A \subseteq B \subseteq V$, với mọi $e \notin B$ ta có lợi nhuận biên của e khi đóng góp vào tập nhỏ hơn, A, sẽ ít nhất bằng lợi nhuận biên khi đóng góp e vào tập lớn hơn, B:

$$f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B).$$

Một lớp bài toán con quan trọng khi nghiên cứu về hàm submodular đó là hàm có *tính chất đơn điệu tăng*, nói ngắn gọn là tính chất đơn điệu (monotone). Hàm đơn điệu nói rằng khi tăng kích cỡ của tập đối số thì giá trị của hàm không bị giảm đi.

Định nghĩa 1.4. (Tính chất đơn điệu) Hàm $f : 2^V \mapsto \mathbb{R}_+$ là hàm submodular đơn điệu nếu với mọi tập $A \subseteq B \subseteq V$ ta có $f(A) \leq f(B)$.

1.2. Lợi ích và ứng dụng của tối đa hàm submodular

Bài toán tối đa hàm submodular, SM, thường được áp dụng khi cần thu thập thông tin nhiều nhất có thể, hay khi ai đó muốn khuyến khích tính đa dạng, tính độc lập, có sự lan truyền hoặc phân tán thông tin **?**.

Sự linh hoạt của hàm submodular làm cho thông tin có thể được thu thập và xử lý hiệu quả. Tóm lại, bài toán SM mang lại các lợi ích trong ứng dụng như sau:

- Hàm submodular có thể hoạt động giống như một dạng hàm thông tin. Ví dụ: hàm thông tin phụ thuộc lẫn nhau... Vì thế, bài toán SM có thể mô hình hóa cho các bài toán cần thu thập thông tin hỗ trợ công tác dự báo và ra quyết định như đặt cảm biến, xây dựng các hệ thống gợi ý, lan truyền thông tin...;

- Cực đại hàm submodular tạo ra sự đa dạng của thông tin. Xu hướng của hàm submodular khi chọn một phần tử mới thêm vào tập lời giải là nó sẽ chọn phần tử nào khác biệt nhất so với các phần tử đã chọn. Hệ quả là *bài toán SM sẽ chọn ra tập con không dư thừa và vì vậy không lãng phí*;

- Bài toán SM sẽ hạn chế sự mất mát thông tin, giúp giảm được kích thước dữ liệu khi có thể tìm ra một tập $S \subseteq V$ có thể đại diện cho toàn bộ tập V.

Từ các lợi ích nói trên, cho thấy ứng dụng bài toán SM có ý nghĩa thiết thực. Đồng thời, bài toán này được vận dụng vào rất nhiều bài toán có giá trị thực tiễn như: Tóm tắt dữ liệu, tối đa ảnh hưởng trên mạng xã hội, tối đa hoá doanh thu; đặt cảm biến tối đa thông tin thu được...

1.3. Các vấn đề nghiên cứu có liên quan

1.3.1. Bài toán tối ưu hàm submodular có ràng buộc

Các nghiên cứu về SM theo nhiều hướng khác nhau. Một số công bố nghiên cứu bài toán SM không có ràng buộc ($\mathcal{C} = \emptyset$)... Các nghiên cứu chỉ ra thuật toán xấp xỉ cho tỉ lệ tốt nhất hiện nay là trả lại tập S sao cho tỉ lệ của lời giải so với lời giải tối ưu S^* thỏa mãn:$f(S) \geq \frac{1}{2} f(S^*)$.

Tuy nhiên, nghiên cứu về SM được tập trung nhiều hơn khi xét bài toán với ràng buộc. Việc đưa thêm ràng buộc vào cũng thỏa mãn nhiều kịch bản thực tế như thời gian, chi phí, số lượng... hầu hết sẽ bị giới hạn. Đối với SM có ràng buộc, đây vẫn là bài toán NP-khó, khả năng mở rộng của Bài toán 1.1 bị hạn chế bởi độ khó của nó.

Vì vậy, các nhà nghiên thường tập trung vào đề xuất các thuật toán xấp xỉ hiệu quả với các đảm bảo lý thuyết về tỉ lệ xấp xỉ để giải quyết bài toán này. Một số ràng buộc tiêu biểu:

- Bài toán SM với ràng buộc lực lượng - SMC. Bài toán yêu cầu tìm tập lời giải $S \subseteq V, |S| \leq k$. Nemhauser và cộng sự là những người đầu tiên đề xuất thuật toán tham lam xấp xỉ cho bài toán SMC có hàm mục tiêu đơn điệu. Thuật toán của họ được chỉ ra tỉ lệ xấp xỉ là $1 - 1/e$ lời giải tối ưu. Họ cũng chứng minh được rằng tính toán hàm f với không gian tập hợp là đa thức sẽ không thể cho tỉ lệ xấp xỉ tốt hơn $(1 - 1/e)$. Sau này, các nhà khoa học tập trung giải bài toán SMC bằng các thuật toán cải tiến giảm thời gian chạy hoặc giảm dung lượng lưu trữ...

- Bài toán SM với ràng buộc chi phí - SMK. Bài toán cho một ngân sách B, yêu cầu tìm lời giải S sao cho tối đa hàm submodular $f(S)$ với ràng buộc chi phí lời giải $c(S) \leq B$. Khi $c(e) = 1, \forall e \in V$, bài toán trở thành tối đa hàm submodular với ràng buộc lực lượng. Do đó, ràng buộc này tổng quát hơn ràng buộc lực lượng. Wolsey và cộng sự khởi xướng nghiên cứu bài toán SMK, họ chứng minh được bài toán này là NP-khó nhưng có thể xấp xỉ lời giải với tỉ lệ $(1 - 1/e)$. Các thuật toán được đề xuất giải các bài toán trên thường được xây dựng dựa trên giải thuật tham lam. Tuy nhiên tối ưu hàm submodular là các bài toán NP-khó, nên các nghiên cứu sau này cải tiến tham lam bằng các thuật toán khác như thuật toán luồng hoặc thuật toán song song...

- Bài toán Phủ Submodular - SC. Cho một hàm submodular đơn điệu không âm $f : 2^V \mapsto \mathbb{R}_+$ và một ngưỡng $\alpha > 0$, bài toán cần tìm một tập $S \subseteq V$ với lực lượng nhỏ nhất sao cho $f(S) \geq \alpha$. Bài toán SC

cũng đã được chứng minh là bài toán NP-khó bởi Wolsey. Ông cũng đồng thời đưa ra một thuật toán tham lam cho tỉ lệ xấp xỉ là $1 - \ln(\max_{e \in V} f(e)/\beta)$ với β là lợi ích nhỏ nhất khác 0 của một phần tử nào đó được thêm vào tập lời giải bởi thuật toán. Sau này, một số tác giả đã đưa các cải tiến vào trong thuật toán để giảm số lượng truy vấn xuống...

1.3.2. Sự mở rộng của bài toán tối ưu hàm submodular

1.3.2.1. Mở rộng hàm mục tiêu thành k-submodular

Các ứng dụng cụ thể của bài toán tối đa hàm k-submodular như tối đa ảnh hưởng k chủ đề, phân lớp, đặt k loại cảm biến... cho thấy việc nghiên cứu về tối ưu hàm k-submodular không chỉ có giá trị lý thuyết mà còn có giá trị thực tiễn.

Bài toán tối đa hàm submodular được mở rộng bài toán tối đa hàm k-submodular với nhiều điều kiện khác nhau như tối đa hàm bi-submoudlar, k-submodular không có ràng buộc, và có ràng buộc... Hàm k-submodular được định nghĩa như sau:

Định nghĩa 1.5 (k-submodular)**.** Cho một tập cơ sở V và một số nguyên dương k, quy ước $[k] = \{1, 2, \ldots, k\}$ và $(k+1)^V = \{(V_1, V_2, \ldots, V_k) | V_i \subseteq V, \forall i \in [k], V_i \cap V_j = \emptyset, \forall i \neq j\}$ là họ k tập không giao nhau được gọi là *k-tập (k-set)*. Hàm $f : (k+1)^V \mapsto \mathbb{R}_+$ là *k-submodular* khi và chỉ khi với các biến $\mathbf{x}, \mathbf{y}$ bất kì $\mathbf{x} = (X_1, X_2, \ldots, X_k)$ và $\mathbf{y} = (Y_1, Y_2, \ldots, Y_k) \in (k+1)^V$, ta có:

$$f(\mathbf{x}) + f(\mathbf{y}) \geq f(\mathbf{x} \sqcap \mathbf{y}) + f(\mathbf{x} \sqcup \mathbf{y}),$$

trong đó: $\mathbf{x} \sqcap \mathbf{y} = (X_1 \cap Y_1, \ldots, X_k \cap Y_k)$, và $\mathbf{x} \sqcup \mathbf{y} = (Z_1, \ldots, Z_k),$ với $Z_i = X_i \cup Y_i \setminus (\bigcup_{j \neq i} X_j \cup Y_j)$.

1.3.2.2. Mở rộng hàm mục tiêu trên lưới nguyên

Các ứng dụng được triển khai trên hàm tập hợp submodular chưa xét đến một tình huống thực tế, một phần tử tốt có thể được lựa chọn nhiều lần vào tập lời giải. Tình huống như vậy thúc đẩy các nhà nghiên cứu tìm hiểu các phiên bản tổng quát hóa của tính submodular và lợi nhuận hiệu suất giảm dần được định nghĩa trên *lưới nguyên (Integer lattice).* Soma và Yoshida lần đầu tiên mở rộng hàm f trên lưới nguyên $\mathbb{Z}_+^V$. Các tác giả đã chỉ ra *tính chất lợi nhuận hiệu suất giảm dần trên lưới nguyên* thường gọi là *DR-submodular)* trở thành phiên bản tổng quát của hàm submodular trên lưới nguyên. Hàm DR-submodular được định nghĩa như sau:

Định nghĩa 1.6 (DR-submodular)**.** Cho một số nguyên dương $k \in \mathbb{N}$, ký hiệu $[k]$ đại diện cho tập $\{1, \ldots, k\}$. Cho tập cơ sở $V = \{e_1, \ldots, e_n\}$, ký hiệu $\mathbf{x}(e)$ là giá trị tọa độ của vec-tơ $\mathbf{x} \in \mathbb{Z}_+^V$ ứng với phần tử e nào đó. Ta cũng ký hiệu vec-tơ đơn vị thứ e là χ_e với $\chi_e(t) = 1$ nếu $t = e$ và $\chi_e(t) = 0$ nếu $t \neq e$. Với mọi vec-tơ $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_+^V$, nói rằng $\mathbf{x} \leq \mathbf{y}$ nếu và chỉ nếu $\mathbf{x}(e) \leq \mathbf{y}(e), \forall e \in V$.

Hàm $f : \mathbb{Z}_+^V \mapsto \mathbb{R}_+$ thỏa mãn tính chất DR-submodular khi và chỉ khi, với $\mathbf{x}, \mathbf{y} \in \mathbb{Z}_+^V, \mathbf{x} \leq \mathbf{y}$:

$$f(\mathbf{x} + \chi_e) - f(\mathbf{x}) \geq f(\mathbf{y} + \chi_e) - f(\mathbf{y}).$$

1.4. Thuật toán xấp xỉ giải quyết bài toán tối ưu hàm submodular

1.4.1. Khái niệm thuật toán xấp xỉ

Giả sử có hàm mục tiêu $f : 2^V \mapsto \mathbb{R}$ với V là tập cơ sở kích thước n. Gọi $S^* \subseteq V$ là *lời giải tối ưu của bài toán (optimal solution)*, giá trị $f(S^*)$ tương ứng là *giá trị tối ưu của bài toán (optimal value)*, ký hiệu là opt. Khi đó, định nghĩa thuật toán xấp xỉ của bài toán CO như sau:

Định nghĩa 1.7 (Thuật toán xấp xỉ)**.** Giả sử cần tìm lời giải $S \subseteq V$ với V là tập cơ sở kích thước n sao cho hàm f đạt giá trị cực đại. Ta nói thuật toán xấp xỉ $\mathcal{A}$ cho lời giải là $S \subseteq V$ có tỉ lệ xấp xỉ là $\rho, \rho \in (0, 1)$, nếu nó thực hiện trong thời gian đa thức theo kích cỡ của dữ liệu đầu vào và thỏa mãn

$$f(S) \geq \rho \cdot \mathsf{opt}. \tag{1.1}$$

Với bài toán tìm giá trị cực tiểu của hàm f, một cách tương tự sẽ có: $f(S) \leq \rho.\mathsf{opt}, \rho > 1$. ρ được gọi là *tỉ lệ xấp xỉ hoặc hệ số xấp xỉ.*

1.4.2. Các đảm bảo lý thuyết của thuật toán

Các tiêu chí thường gặp để đánh giá một thuật toán và so sánh giữa các thuật toán:

- Tỉ lệ xấp xỉ đảm bảo chất lượng lời giải. Tỉ lệ xấp xỉ ρ có ý nghĩa rằng trong trường hợp xấu nhất, thuật toán cũng đảm bảo đạt tỉ lệ ρ lần lời giải tối ưu.

- Khi giải các bài toán cần tối ưu chi phí, nhiều nghiên cứu xây dựng các thuật toán xấp xỉ tiêu chí kép. Một thuật toán gọi là xấp xỉ tiêu chí kép (σ_1, σ_2) là thuật toán cho 2 tỉ lệ xấp xỉ σ_1, σ_2 lần lượt là các tỉ lệ xấp xỉ của giá trị chi phí và giá trị mục tiêu của lời giải có được từ thuật toán so với lời giải tối ưu.

- Độ phức tạp thời gian. Thông thường, độ phức tạp xấu nhất (hoặc tồi nhất) được sử dụng thường xuyên hơn cả để đánh giá độ phức tạp thuật toán, vì nó đánh giá khả năng xấu nhất sẽ xảy ra của một thuật toán, để người sử dụng có thể ước lượng được thời gian hoặc chi phí cần thiết để thực hiện thuật toán.

Để làm căn cứ đánh giá một cách tổng quát các thuật toán, người ta thường *xấp xỉ thời gian chạy của thuật toán*. Trong đó, hàm O, còn gọi là O-lớn (*Big-oh*) cho một xấp xỉ của thời gian chạy của thuật toán trong trường hợp xấu nhất.

- Độ phức tạp truy vấn. Giả sử bài toán đã cho một cách ước lượng hàm mục tiêu f, coi một lời gọi hàm là một phép tính đa thức theo thời gian, một truy vấn là một lời gọi hàm f. Độ phức tạp truy vấn là số lượng truy vấn nhiều nhất mà thuật toán cần thực hiện.

Với các bài toán CO nói chung và các bài toán tối ưu hàm submodular nói riêng, việc gọi hàm f là thường xuyên, nên số lượng lời gọi hàm ảnh hưởng trực tiếp đến thời gian chạy của thuật toán. Do vậy đây là một phép đo quan trọng để xác định hiệu quả của một thuật toán. Các thuật toán thiết kế theo hướng giảm độ phức tạp truy vấn đang là xu hướng chung khi giải bài toán tối ưu hàm submodular.

1.4.3. Thuật toán tham lam xấp xỉ

Thuật toán tham lam xấp xỉ là một thuật toán hiệu quả để giải quyết các bài toán tối ưu hàm submodular. Nemhauser và Wolsey cũng đã chỉ ra sự phù hợp của thuật toán tham lam với bài toán SM khi cho tỉ lệ xấp xỉ tốt nhất.

Nhược điểm của giải thuật tham lam đó là số phép tính cần thực hiện rất lớn. Thuật toán chạy tuần tự, tại mỗi bước lặp cần tốn $O(n)$ truy vấn hàm f. Như vậy, Thuật toán tham lam giải quyết bài toán SMC

chẳng hạn có độ phức tạp thời gian là $O(nk)$, độ phức tạp truy vấn là $O(nk)$, độ phức tạp bộ nhớ là $O(n)$. Điều này làm thuật toán trở nên bất khả thi với bộ dữ liệu có kích cỡ lớn.

Để khắc phục vấn đề thời gian chạy của thuật toán tham lam, các nhà khoa học thường đưa ra các giải pháp cải tiến theo hướng giảm độ phức tạp thời gian, giảm độ phức tạp truy vấn, hoặc giảm độ phức tạp bộ nhớ nhưng vẫn đảm bảo tỉ lệ xấp xỉ chấp nhận được. Một số phương pháp cải tiến phổ biến có thể kể đến là: Thiết kế ngưỡng tham lam; Thiết kế thuật toán luồng; Thiết kế thuật toán song song...

CHƯƠNG 2
BÀI TOÁN TỐI ĐA HÀM k-SUBMODULAR VỚI RÀNG BUỘC CHI PHÍ

Vệc sử dụng mô hình tối ưu hàm submodular chưa đủ để đáp ứng cho một vài tình huống xảy ra khi cần tìm giải pháp để phân lớp các phần tử vào nhiều tập khác nhau nhằm thu thập thông tin với nhiều chủ đề khác nhau. Khi đó, một tập dữ liệu được phân hoạch thành k tập không giao nhau, mỗi một phần tử của tập dữ liệu cần lựa chọn đưa vào tập i nào đó trong k tập con sao cho lượng thông tin thu thập được có lợi nhất. Submodular được mở rộng thành hàm k-submodular. Các bài toán tối ưu hàm k-submodular đã trở thành một chủ đề nghiên cứu thu hút được nhiều sự quan tâm gần đây.

Tiếp cận với xu hướng nghiên cứu này, luận án giải bài toán *tối đa hàm k-submodular với ràng buộc chi phí (k-Submodular Maximization under Knapsack constraint - kSMK)* bằng các thuật toán xấp xỉ. Bài toán kSMK có thể được vận dụng để giải quyết nhiều kịch bản trong thực tiễn như tối đa ảnh hưởng của k chủ đề, đặt k cảm biến, phân lớp... với giới hạn chi phí về nhân lực, ngân sách, thời gian... Tuy nhiên, các nghiên cứu cho kSMK còn chưa nhiều, bộc lộ một số hạn chế với vấn đề thời gian chạy và tỉ lệ xấp xỉ của thuật toán.

Trong chương này, luận án trình bày các thuật toán xấp xỉ tất định cho bài toán kSMK với hai trường hợp, hàm mục tiêu đơn điệu và không đơn điệu.

2.1. Phát biểu bài toán, ứng dụng của bài toán và các thách thức

2.1.1. Phát biểu bài toán

Ta có phát biểu về bài toán tối đa hàm k-submodular với ràng buộc chi phí tương ứng như sau:

Định nghĩa 2.1 (Bài toán kSMK)**.** Với ngân sách giới hạn $B > 0$ cho trước, mỗi phần tử $e \in V$ có một chi phí dương $c(e) \leq B$. Bài toán kSMK yêu cầu tìm một k-tập $\mathbf{s} = (S_1, S_2, \ldots, S_k)$ với tổng chi phí $c(\mathbf{s}) = \sum_{i\in[k]} \sum_{e\in S_i} c(e) \leq B$ sao cho $f(\mathbf{s})$ là cực đại.

Trong đó, V là tập cơ sở, k là một số nguyên dương, quy ước $[k] = \{1, 2, \ldots, k\}$ và $(k+1)^V = \{(V_1, V_2, \ldots, V_k) | V_i \subseteq V, \forall i \in [k], V_i \cap V_j = \emptyset, \forall i \neq j\}$ là họ k tập không giao nhau được gọi là *k-tập (k-set)*.

2.1.2. Ứng dụng của bài toán và các thách thức

Tối đa hàm k-submodular giải quyết nhiều kịch bản mà tối đa hàm submodular chưa thể giải quyết được. Có thể chỉ ra một vài ví dụ như các trường hợp bên dưới. Khi các ứng dụng này bị ràng buộc bởi thời gian, nhân lực hay ngân sách, các ứng dụng trở thành các thể hiện cụ thể của kSMK. Một số ứng dụng tiêu biểu của bài toán này:

- Bài toán tối đa ảnh hưởng theo k chủ đề.
- Bài toán tối đa hóa độ phủ của thông tin theo k chủ đề.

- Bài toán đặt k loại cảm biến.

- Một số ứng dụng khác: Max-k cut; Lựa chọn đặc trưng cho k lớp.

Tuy nhiên, khi giải các bài toán cần phải giải quyết một số vấn đề còn thách thức:

- Vì tối đa hóa hàm submodular là bài toán NP-khó, nên tối đa hóa hàm k-submodular nói chung và bài toán kSMK nói riêng cũng là một bài toán NP-khó.

- Không gian tìm kiếm lời giải tăng theo hàm mũ khi kích cỡ n của tập dữ liệu tăng lên.

- Thách thức đến từ tính chất k-submodular của hàm mục tiêu khác về bản chất với hàm submodular;

- Thách thức từ ràng buộc chi phí.

- Các nghiên cứu về kSMK hiện nay còn chưa nhiều.

- Vấn đề hàm mục tiêu không còn đơn điệu

- Thách thức từ giảm độ phức tạp truy vấn xuống còn tuyến tính.

2.2. Các thuật toán xấp xỉ cho bài toán kSMK đơn điệu tăng

2.2.1. Kết quả mới của luận án

Tổng thể, những đóng góp của luận án đối với bài toán kSMK đơn điệu tăng bao gồm:

- Đề xuất thuật toán xấp xỉ nhanh, **FA** (Fast Approximation), cho tỉ lệ xấp xỉ là $1/10$, cần 1 lần quét tập cơ sở với độ phức tạp truy vấn là $O(kn)$. Thuật toán được xây dựng dựa trên ý tưởng chia đôi tập cơ sở V thành 2 tập nhằm tìm giải pháp tối ưu trên tập thứ nhất và tìm giải pháp gần tối ưu trên tập thứ hai. Đây là thuật toán đơn giản đầu tiên nhưng quan trọng bởi vì nó giới hạn khoảng của giá trị tối ưu và cung cấp một chiến thuật chia tập dữ liệu để giảm độ phức tạp truy vấn còn $O(nk)$.

- Đề xuất thuật toán xấp xỉ nhanh cải tiến, **IFA** (Improved Fast Approximation), cho tỉ lệ xấp xỉ là $1/4 - \epsilon$, yêu cầu độ phức tạp truy vấn là $O(kn/\epsilon)$ với $\epsilon \in (0, 1)$ là tham số chính xác. Thuật toán được xây dựng dựa trên ngưỡng mật độ (density gain threshold) của các phần tử. Thuật toán này cho tỉ lệ xấp xỉ **tốt tương đương** thuật toán DS là một thuật toán tất định tốt nhất hiện nay.

- Chất lượng của **IFA** được cải tiến bằng thuật toán xấp xỉ cải tiến tăng cường, **IFA**+ (Improved Fast Approximation Plus). Thuật toán này cần $O(kn \log(1/\epsilon)/\epsilon)$ truy vấn nhưng có thể cung cấp tỉ lệ xấp xỉ lên tới $(1/3 - \epsilon)$. Thuật toán được xây dựng dựa trên ngưỡng mật độ và tăng cường chất lượng của lời giải dự tuyển bằng cách phân hoạch lại lời giải dự tuyển và bổ sung thêm các phần tử tốt còn trong tập cơ sở. Đây là thuật toán cho **tỉ lệ xấp xỉ tốt hơn** so với thuật toán cho tỉ lệ xấp xỉ tốt nhất hiện nay là thuật toán Greedy đề xuất bởi Tang và cộng sự.

- Để kiểm nghiệm các đóng góp về mặt lý thuyết, luận án thực hiện một số thực nghiệm tổng quát với ba ứng dụng của kSMK, bao gồm: Tối đa ảnh hưởng theo k chủ đề với ràng buộc chi phí - kIMK, tối đa hóa mức độ bao phủ thông tin k chủ đề - kCMK và đặt vị trí k loại cảm biến - kSPK. Kết quả thử nghiệm đã chỉ ra rằng các thuật toán nêu trong luận án không chỉ yêu cầu số lượng truy vấn thấp hơn lên tới hàng trăm lần so với Greedy, thấp hơn tới 15-20 lần so với các thuật toán luồng mà còn cho chất lượng lời giải tốt hơn khoảng 1.5 lần so các thuật toán luồng tốt nhất hiện nay.

Ngoài ra, các thuật toán nêu trong luận án có ưu thế vượt trội hơn các thuật toán còn lại khi B tăng và n tăng. Giá trị đóng góp này cho thấy các thuật toán của luận án có thể làm việc được với các tập dữ liệu có kích thước lớn.

2.2.2. Thuật toán xấp xỉ tăng cường: IFA+

Ý tưởng của thuật toán này dựa trên giảm dần ngưỡng nhằm bổ sung thêm các phần tử tốt từ tập V vào lời giải dự tuyển $\mathbf{s}$. Sau đó, phân hoạch lại $\mathbf{s}$ để lấy thêm các phần tử tốt hơn từ V đưa vào các phân hoạch của $\mathbf{s}$ trong khi tổng chi phí vẫn còn nhỏ hơn ngân sách. Ý tưởng này xuất phát từ nhận xét khi ta chọn được một tập lời giải dự tuyển thì các phần tử tốt vẫn còn nằm ở trong tập V. Do vậy, ta có thể tăng cường chất lượng lời giải của $\mathbf{s}$.

Algorithm 1 IFA+

Input: V, f, k, $B > 0$, $\epsilon > 0$.
Output: Output: A solution $\mathbf{s}$.

\\ Phase 1
$\mathbf{s}_{max} \leftarrow \mathbf{FA}(V, f, k, B)$, $\mathbf{s} \leftarrow \mathbf{0}$;
$\Gamma \leftarrow f(\mathbf{s}_{max})$, $\theta \leftarrow 10\Gamma/(3\epsilon B)$
while $\theta \geq (1-\epsilon)\Gamma/(3B)$ **do**
  for all $e \in V \setminus supp(\mathbf{s})$ **do**
    $i_e \leftarrow \arg\max_{i\in[k]} \Delta_{(e,i)} f(\mathbf{s})$
    if $c(\mathbf{s}) + c(e) \leq B$ **and** $\Delta_{(e,i_e)} f(\mathbf{s})/c(e) \geq \theta$ **then**
      $\mathbf{s} \leftarrow \mathbf{s} \sqcup (e, i_e)$
    end if
  end for
  $\theta \leftarrow (1-\epsilon)\theta$
end while
\\ Phase 2: Boosting quality of solutions
$l \leftarrow \epsilon B$
while $l \leq B$ **do**
  Find $q \leftarrow \max\{i : i \leq |supp(\mathbf{s})|, c(\mathbf{s}^i) \leq l\}$
  $\mathbf{s}'_l \leftarrow \mathbf{s}^q$
  if $\mathbf{s}'_l \neq \mathbf{s}_{(l/(1+\epsilon))}$ **then**
    $(e_l, i_l) \leftarrow \arg\max_{i\in[k], e\in V\setminus supp(\mathbf{s}'_l): c(\mathbf{s}'_l)+c(e)\leq B} \Delta_{(e,i)} f(\mathbf{s}'_l)$
    $\mathbf{s}_{(l)} \leftarrow \mathbf{s}'_l \sqcup (e_l, i_l)$
  end if
  $l \leftarrow (1+\epsilon)l$
end while
$\mathbf{s} \leftarrow \arg\max_{\mathbf{s}''\in\{\mathbf{s}_{max}, \mathbf{s}, \mathbf{s}_{(\epsilon B)}, \mathbf{s}_{(\epsilon B(1+\epsilon))}, \ldots, \mathbf{s}_{(l)}\}} f(\mathbf{s}'')$
return $\mathbf{s}$

Định lý phát biểu về đảm bảo lý thuyết của thuật toán.

Định lý 2.1. *Với $\epsilon \in (0, 1/3)$ bất kỳ, thuật toán* **IFA+** *có độ phức tạp truy vấn là $O(kn\log(1/\epsilon)/\epsilon)$ và trả về lời giải với tỉ lệ xấp xỉ là $1/3 - \epsilon$.*

2.3. Các thuật toán cho trường hợp hàm mục tiêu không đơn điệu

2.3.1. Kết quả mới của luận án

Các bài toán ứng dụng của tối đa hàm submodular, hàm mục tiêu được chỉ ra là có thể không đơn điệu. Hệ quả tất yếu bài toán tối đa hàm k-submodular cũng có thể không đơn điệu. Pham và cộng sự cũng đã giải quyết bài toán tổng quát tối đa hàm mục tiêu k-submodular với ngân sách giới hạn, hàm mục tiêu có thể không đơn điệu. Trong nghiên cứu của họ, các tác giả đưa ra lời giải xấp xỉ với độ phức tạp truy vấn gần tuyến tính. Luận án tập trung giải quyết bài toán với hàm không đơn điệu bằng một thuật toán xấp xỉ cạnh tranh, $\mathbf{RLA}$, và giảm độ phức tạp truy vấn.

Phần thực nghiệm so sánh giữa các thuật toán được đề xuất, $\mathbf{LAA}$ và $\mathbf{RLA}$, của luận án với các thuật toán tốt nhất hiện nay, thuật toán luồng tất định (DS) và thuật toán luồng ngẫu nhiên (RS) đã đề cập đến ở phần trước, theo ba khía cạnh: tỉ lệ xấp xỉ, độ phức tạp của truy vấn và tính tất định hay không. Thuật toán $\mathbf{LAA}$ là phiên bản đơn giản, có tác dụng đưa số truy vấn về tuyến tính và giới hạn lại khoảng giá trị của opt để $\mathbf{RLA}$ sử dụng. Các kết quả cho thấy thuật toán được đề xuất có số lượng truy vấn thấp hơn và tỷ lệ xấp xỉ hằng số có giá trị tốt tương đương các thuật toán hiện đại.

2.3.2. Thuật toán tuyến tính cải tiến: RLA

$\mathbf{RLA}$ giữ ý tưởng của $\mathbf{IFA}$ bằng cách sử dụng lại giải pháp của $\mathbf{LAA}$ để giới hạn phạm vi của opt và điều chỉnh ngưỡng tham lam để cải thiện tỷ lệ xấp xỉ bằng cách tiến hành $O(1/\epsilon)$ lần quét qua tập cơ sở. Chi tiết thuật toán được trình bày đầy đủ trong Thuật toán 2.

Algorithm 2 $\mathbf{RLA}$

Input: V, f, k, $B > 0$, $\epsilon > 0$.
Output: A solution $\mathbf{s}$

$\mathbf{s}_b \leftarrow \mathbf{LAA}(V, f, k, B)$
$\Gamma \leftarrow f(\mathbf{s}_b)$
$A \leftarrow \{(1+\epsilon)^i : i \in \mathbb{N}, \Gamma \leq (1+\epsilon)^i \leq 19\Gamma\}$
for all $e \in V$ **do**
  for all $v \in A$ **do**
    $i_v \leftarrow \arg\max_{i \in [k]} \Delta_{(e,i)} f(\mathbf{s}_v)$
    $\tau_v = 2v/(5B)$
    if $c(\mathbf{s}_v) + c(e) \leq B$ **and** $\Delta_{(e,i_v)} f(\mathbf{s}_v)/c(e)) \geq \tau_v$ **then**
      $\mathbf{s}_v \leftarrow \mathbf{s}_v \sqcup (e, i_v)$
    end if
  end for
end for
$\mathbf{s}_{final} \leftarrow \arg\max_{\mathbf{s}' \in \{\mathbf{s}_{max}, \mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{|S|}\}} f(\mathbf{s}')$
return $\mathbf{s}_{final}$

Định lý 2.2. *Với $0 < \epsilon < 1/5$, thuật toán $\mathbf{RLA}$ trả về tỉ lệ xấp xỉ là $1/5 - \epsilon$, trong với độ phức tạp truy vấn là $O(nk/\epsilon)$.*

Ngoài ra, luận án cũng tiến hành thực nghiệm trên một số bài toán ứng dụng của kSMK. Các bài toán ứng dụng này là các trường hợp của kSMK thể hiện cụ thể nó được áp dụng vào đâu. Đây đều là các bài toán phổ biến và có tính thực tiễn, bao gồm :

- Tối đa ảnh hưởng của k chủ đề trong giới hạn chi phí (k-topic Influence Maximization under Knapsack constraint - kIMK);

- Tối đa độ phủ thông tin của k chủ đề trong giới hạn chi phí (k-topic information Coverage Maximization under Knapsack constraint - kCMK);

- Tối ưu vị trí đặt k loại cảm biến trong giới hạn chi phí (k-type Sensor Placement under Knapsack constraint - kSPK).

Phần thực nghiệm so sánh giữa các thuật toán cho thấy sự tiệm cận về lý thuyết với thực tiễn, củng cố kết luận các thuật toán do luận án đóng góp cải tiến đáng kể các đảm bảo lý thuyết.

CHƯƠNG 3
BÀI TOÁN TỐI ĐA HÀM SUBMODULAR ĐƠN ĐIỆU VỚI RÀNG BUỘC CHI PHÍ CÓ NHIỄU

Ở chương trước, luận án đã tiếp cận hướng nghiên cứu mở rộng hàm mục tiêu từ submodular sang k-submodular với nhiều ý nghĩa lý thuyết và thực tiễn. Tuy nhiên khi nghiên cứu về các bài toán này, rất ít tác giả đặt vấn đề ước lượng hàm mục tiêu bị chi phối bởi nhiễu. Trong khi đó, vấn đề với nhiễu là một vấn đề thường gặp và tiêu tốn rất nhiều chi phí để xử lý nó.

Đặt vấn đề nghiên cứu với nhiễu, luận án giải bài toán *tối đa hàm submodular với ràng buộc chi phí có nhiễu (Submodular Maximization subject to a Knapsack constraint under Noises - SMKN)*. Đây là một bài toán mới, hiện chưa có nghiên cứu nào giải quyết bài toán này. Thêm vào đó, độ khó của bài toán NP-khó và ràng buộc chi phí làm cho bài toán có tính thách thức hơn.

3.1. Phát biểu bài toán và các thách thức của bài toán

3.1.1. Phát biểu bài toán

Luận án đặt vấn đề xem xét bài toán SMK dưới sự tác động của nhiễu được mô hình hoá thành bài toán SMKN. Hai loại nhiễu được xem xét là *nhiễu cộng (additive noise)* và *nhiễu nhân (multiplicative noise)*. Bài toán này có định nghĩa như sau:

Định nghĩa 3.1 (Bài toán SMKN)**.** Bài toán SMK có nhiễu tồn tại một hàm F là ước lượng nhiễu của hàm mục tiêu f submodular đơn điệu. Gọi $\epsilon > 0$ là tham số nhiễu, hàm f dưới sự ảnh hưởng của nhiễu cộng hoặc nhiễu nhân được xấp xỉ theo ước lượng như sau:

- Với **nhiễu nhân** ϵ, có:

$$(1-\epsilon)f(S) \leq F(S) \leq (1+\epsilon)f(S). \tag{3.1}$$

- Với **nhiễu cộng** ϵ, có:

$$f(S) - \epsilon \leq F(S) \leq f(S) + \epsilon. \tag{3.2}$$

3.1.2. Các thách thức của bài toán

Với bài toán SMKN, đây là một bài toán mới, việc đề xuất các thuật toán xấp xỉ hiệu quả nhằm giải quyết bài toán cần phải đối mặt với nhiều thách thức, có thể kể đến như sau:

- Thách thức của ràng buộc chi phí.
- Thách thức về khả năng mở rộng của thuật toán khi dữ liệu đầu vào tăng lên.
- Thách thức khi xử lý bài toán SMK với nhiễu, vì hàm F có thể không thừa kế lại các tính chất của hàm f.
- Thách thức về xây dựng ước lượng F của hàm mục tiêu phù hợp với từng bài toán ứng dụng cụ thể.

Tóm lại, chưa có tác giả nào đặt vấn đề nghiên cứu bài toán SMK với nhiễu và đề xuất thuật toán khả thi để giải. Do vậy, luận án giải bài toán *tối đa hàm submodular với ràng buộc chi phí có nhiễu, SMKN,* bằng các thuật toán xấp xỉ hiệu quả. Các thách thức trên đây vừa là khó khăn vừa là động lực để luận án tiến hành nghiên cứu bài toán SMKN theo các tiêu chí: giảm thời gian chạy của thuật toán, giảm dung lượng lưu trữ cần cho các phần tử và cải thiện tỉ lệ xấp xỉ có thể cạnh tranh được với các thuật toán tham lam.

3.2. Thuật toán xấp xỉ cho bài toán SMKN

3.2.1. Kết quả mới của luận án

Đầu tiên, luận án đề xuất một phiên bản thuật toán của tham lam xấp xỉ với điều kiện đã biết ước lượng xấp xỉ F của hàm mục tiêu. Thuật toán tiếp theo là phiên bản cải tiến được thiết kế trở thành thuật toán luồng khi biết ước lượng xấp xỉ F của hàm mục tiêu. Mục đích của phiên bản đầu tiên là thiết kế một thuật toán tham lam làm cơ sở ban đầu cho bài toán SMKN. Phiên bản thứ hai cải tiến thuật toán tham lam, giảm thời gian tính toán và dung lượng lưu trữ với chất lượng lời giải vẫn được đảm bảo. Vì SMKN là một bài toán mới, nên phần thực nghiệm sẽ trực tiếp so sánh và đánh giá hai thuật toán này.

Thực hiện so sánh **GUN** và **NS** theo 3 khía cạnh: chất lượng lời giải, độ phức tạp truy vấn, và độ phức tạp bộ nhớ, hai thuật toán này cho chất lượng lời giải trong môi trường nhiễu là không chênh lệch quá lớn. Với **GUN**, các độ phức tạp truy vấn và bộ nhớ là $O(n^2)$, trong khi với **NS**, độ phức tạp bộ nhớ ít hơn so với độ phức tạp truy vấn một hệ số là n. Bộ nhớ trong **NS** phụ thuộc chủ yếu vào ngân sách B.

3.2.2. Thuật toán tham lam xấp xỉ: GUN

Với bài toán SMKN, đầu tiên, ta thiết kế một thuật toán tham lam dành cho nó. Luận án giới thiệu phiên bản thuật toán tham lam, **GUN** (*Greedy Under Noises*), được thiết kế dựa trên ý tưởng chọn các phần tử cho tỉ lệ đóng góp lợi nhuận biên và chi phí của phần tử đó là lớn nhất trong các phần tử chưa được chọn.

Algorithm 3 **GUN**

Input: V, an approximation oracle F, $\epsilon \in (0, 1)$, budget $B > 0$
Output: A subset S

$S \leftarrow \emptyset, U \leftarrow V$
if F: ϵ-multiplicative noise oracle **then**
  for $U = \emptyset$ **do**
    $e' \leftarrow \arg\max_{e \in U} \frac{1}{c(e)} \left(\frac{F(S+e)}{1-\epsilon} - \frac{F(S)}{1+\epsilon} \right)$
    if $c(S + e') \leq B$ **then**
      $S \leftarrow S + e'$
    end if
    $U \leftarrow U \setminus \{e'\}$
  end for
end if
if F: ϵ-additive noise oracle **then**
  for $U = \emptyset$ **do**
    $e' \leftarrow \arg\max_{e \in U} \frac{1}{c(e)} \left(F(S + e) - F(S) \right)$
    if $c(S + e') \leq B$ **then**
      $S \leftarrow S + e'$
    end if
    $U \leftarrow U \setminus \{e'\}$
  end for
end if
$e_m \leftarrow \arg\max_{e \in V} F(e)$
$S \leftarrow \arg\max_{S' \in \{S, e_m\}} F(S')$
return S

Chất lượng lời giải của thuật toán trên được thể hiện qua Định lý 3.1:

Định lý 3.1. *Thuật toán* **GUN** *cho lời giải S đảm bảo:*

$$f(S) \geq \frac{1}{2}\frac{1-\epsilon}{1+\epsilon}\left(1-\frac{1}{e}\right)\left(1-\frac{4\epsilon B}{1-\epsilon^2}\right)\mathsf{opt}$$

cho nhiễu nhân ϵ và

$$f(S) \geq \frac{1}{2}\left(1-\frac{1}{e}\right)\mathsf{opt} - 2\epsilon(B+1)$$

cho nhiễu cộng ϵ.

Thuật toán này có độ phức tạp là $O(n^2)$ (về thời gian chạy và bộ nhớ cần sử dụng, vì thuật toán là tuần tự).

3.2.3. Thuật toán xấp xỉ tổng quát: NS

Để cải tiến thuật toán tham lam, luận án đề xuất một thuật toán luồng 1 lần quét (tức là duyệt tập cơ sở 1 lần) cho bài toán SMKN. **GUN** phải quét tập cơ sở nhiều lần để chọn ra phần tử cho tỉ lệ lợi nhuận biên và chi phí lớn nhất. Do đó, làm tốn thời gian chạy. Sử dụng kỹ thuật luồng, tại một thời điểm ta chỉ quét 1 lượng phần tử nhỏ so với toàn tập cơ sở và lưu trong bộ nhớ. Như vậy, thời gian và dung lượng bộ nhớ đều được giảm xuống.

Ta xây dựng được định lý phân tích đảm bảo lý thuyết sau:

Định lý 3.2. *Với nhiễu nhân ϵ, thuật toán* **NS** *là thuật toán luồng 1 lần quét cho độ phức tạp truy vấn $O(\frac{n}{\gamma}\log(B\frac{1+\epsilon}{1-\epsilon}))$, độ phức tạp bộ nhớ $O(\frac{B}{\gamma}\log(B\frac{1+\epsilon}{1-\epsilon}))$ và có tỉ lệ xấp xỉ là $(\frac{1-\epsilon}{1+\epsilon})^2\frac{(1-\gamma)}{(\frac{1-\epsilon}{1+\epsilon})^2+2(\frac{1}{1-\epsilon}+B\frac{3\epsilon-\epsilon^2}{1-\epsilon^2})}$.*

Tương tự với Định lý 3.2, đảm bảo chất lượng lời giải của Thuật toán 4 được khẳng định bằng định lý dưới đây:

Định lý 3.3. *Dưới tác động của nhiễu cộng ϵ, thuật toán* **NS** *là thuật toán luồng 1 lần quét với độ phức tạp truy vấn là $O(\frac{n}{\gamma}\log(\frac{mB+\epsilon}{m-\epsilon}))$, độ phức tạp bộ nhớ là $O(\frac{B}{\gamma}\log(\frac{mB+\epsilon}{m-\epsilon}))$ và trả về kết quả S_{str} thỏa mãn $f(S_{str}) \geq \frac{1-\gamma}{3}\mathsf{opt} + \epsilon - 2\epsilon B$, với $m = \max_{e\in V} F(e)$.*

Để cụ thể hóa các nghiên cứu lý thuyết, luận án đưa ra một số thực nghiệm để so sánh các thuật toán đã đề xuất cho một trường hợp cụ thể của SMKN, đó là xét ước lượng nhiễu trên *tối đa ảnh hưởng với ràng buộc chi phí (Influence Maximization under Knapsack constraint- IMK)*. Phần kết quả thực nghiệm cho thấy về chất lượng lời giải **GUN** cho kết quả tốt nhất, nhưng **NS** thấp hơn không đáng kể. Ngược lại, **NS** lại cho thấy sự vượt trội khi giảm số lượng truy vấn, thời gian chạy và bộ nhớ dùng cho tập dữ liệu khi chạy thuật toán.

CHƯƠNG 4
BÀI TOÁN PHỦ SUBMODULAR ĐƠN ĐIỆU TRÊN LƯỚI NGUYÊN

Chương 2 luận án đã nghiên cứu vấn đề mở rộng hàm submodular thành k-submodular đáp ứng một số tình huống khi cần phân lớp các phần tử thành các tập không giao nhau, phục vụ một số tình huống thực tiễn như tối đa ảnh hưởng k-chủ đề, đặt k loại cảm biến... Tuy nhiên, trong thực tế còn tồn tại các tình huống một phần tử tốt *có thể được lựa chọn nhiều lần* vào tập lời giải. Ví dụ khi một công ty chọn các đại lý tốt để tiếp thị sản phẩm, công ty sẽ có xu hướng chọn đi chọn lại các đại lý mang lại lợi nhuận cao. Các tình huống như vậy cần mở rộng hàm submodular trên lưới nguyên.

Algorithm 4 : **NS**

Input: V, F,$\epsilon \in (0,1)$,B, $\gamma \in (0,1)$
Output: A solution S

$m \leftarrow 0, O = \{(1+\gamma)^i | i \in \mathbb{Z}_+\}$
for all $v \in O$ **do**
  $S_v = \emptyset$
end for
if f: an ϵ-multiplicative noise oracle **then**
  $\alpha = \frac{2}{\frac{1-\epsilon}{1+\epsilon} + \frac{2}{1-\epsilon} + 2B\frac{3\epsilon-\epsilon^2}{1-\epsilon^2}}$
  for all $e \in V$ **do**
    $e_m \leftarrow \arg\max_{e' \in \{e_m, e\}} F(e'), m \leftarrow f(e_m)$
    $O = \{(1+\gamma)^i | \frac{m}{1+\epsilon} \leq (1+\gamma)^i \leq \frac{Bm}{1-\epsilon}, i \in \mathbb{Z}_+\}$
    Delete S_v for each $v \notin O$
    for all $v \in O$ and $c(S_v + e) \leq B$ **do**
      if $\frac{F(S_v+e)}{1-\epsilon} \geq \frac{\alpha v c(S_v+e)}{B}$ **then**
        $S_v \leftarrow S_v + \{e\}$
      end if
    end for
  end for
else if f: ϵ-additive noise oracle **then**
  for all $e \in V$ **do**
    $e_m \leftarrow \arg\max_{e' \in \{e_m, e\}} F(e'), m \leftarrow f(e_m)$
    $O = \{(1+\gamma)^i | m - \epsilon \leq (1+\gamma)^i \leq B(m+\epsilon), i \in \mathbb{Z}_+\}$
    Delete S_v for each $v \notin O$
    for all $v \in O$ and $c(S_v + e) \leq B$ **do**
      $\alpha_v \leftarrow \frac{2}{3} + \frac{2\epsilon}{v} - \frac{4\epsilon B}{3v}$
      if $F(S_v + e) + \epsilon \geq \frac{\alpha v c(S_v+e)}{B}$ **then**
        $S_v \leftarrow S_v + \{e\}$
      end if
    end for
  end for
end if
// Binary Search on S_v
Find $S_{str} \leftarrow \arg\max_{S \in \{S_v : v \in O\} \cup \{e_m\}} F(S_v)$ by BS
return S_{str}

Luận án đặt vấn đề nghiên cứu bài toán Phủ Submodular (Submodular Cover - SC) đơn điệu trên lưới nguyên vì đây là một bài toán mới. Các nghiên cứu hiện nay còn một số hạn chế khi hàm mục tiêu cho giá trị nguyên hoặc độ phức tạp truy vấn cao. Luận án giải quyết bài toán bằng thuật toán xấp xỉ tiêu chí kép được thiết kế song song với độ phức tạp truy vấn và độ phức tạp song song thấp, khắc phục các hạn chế của các nghiên cứu hiện có.

4.1. Phát biểu bài toán, ứng dụng và các thách thức của bài toán

4.1.1. Phát biểu bài toán

Định nghĩa 4.1 (Bài toán SC)**.** Cho một hàm submodular đơn điệu không âm $f : 2^V \mapsto \mathbb{R}_+$ và một ngưỡng $\alpha > 0$, bài toán yêu cầu tìm một tập lời giải $S \subseteq V$ với lực lượng nhỏ nhất sao cho $f(S) \geq \alpha$.

Trong chương này, luận án mở rộng nghiên cứu giải bài toán SC trên lưới nguyên, với hàm mục tiêu submodular được mở rộng thành hàm *DR-submodular (Disminishing Return Submodular)*. Bài toán SC

được tổng quát hoá thành Phủ DR-Submodular (DRSC) được phát biểu như sau:

Định nghĩa 4.2 (Bài toán DRSC)**.** Cho một hàm DR-submodular đơn điệu $f : \mathbb{Z}_+^V \mapsto \mathbb{R}_+$, một số nguyên dương B là giá trị lớn nhất trên một trục tọa độ bất kỳ của một vec-tơ trong $\mathbb{Z}_+^V$, và một ngưỡng $\alpha > 0$. Bài toán cần tìm vec-tơ lời giải $\mathbf{x} \leq \mathbf{B}$ có lực lượng nhỏ nhất sao cho hàm mục tiêu không nhỏ hơn α, hay:

$$\min \ \|\mathbf{x}\|_1 \quad \text{s.t} \quad f(\mathbf{x}) \geq \alpha, \mathbf{0} \leq \mathbf{x} \leq \mathbf{B}, \tag{4.1}$$

trong đó $\mathbf{B} = B \cdot \mathbf{1}$ và $\|\mathbf{x}\|_1 = \sum_{e \in V} \mathbf{x}(e)$.

Đối với hàm submodular là hàm tập hợp, một phần tử e của tập cơ sở chỉ được chọn một lần. Để phần tử e được chọn lại nhiều lần cần có sự mở rộng tập hợp trên lưới nguyên. Khi đó, tập hợp trở thành đa tập hợp $\{\mathbf{x}\}$, tương ứng với một vec-tơ $\mathbf{x} \in \mathbb{Z}_+^V$. Số lần phần tử e xuất hiện gọi là $\mathbf{x}(e)$. Ví dụ, cho tập cơ sở $V = \{e_1, e_2, e_3\}$, vec-tơ $\mathbf{x} = (2, 4, 0)$ trên lưới nguyên biểu diễn đa tập hợp $\{\mathbf{x}\}$ tương ứng $\{e_{11}, e_{12}, e_{21}, e_{22}, e_{23}, e_{24}, e_{21}\}$. Với một tập con $A \subseteq V$ bất kỳ, kích cỡ của $\mathbf{x}$ là $\|\mathbf{x}\|_1 = \sum_{e \in A} \mathbf{x}(e)$, là tổng số phần tử có mặt trong đa tập hợp $\{\mathbf{x}\}$ tương ứng, trong ví dụ trên là 7 phần tử.

4.1.2. Ứng dụng của bài toán và các thách thức của bài toán

Bài toán có ứng dụng khi một phần tử cho lợi ích "tốt" được lựa chọn nhiều lần. Một số tình huống thực tiễn được đặt ra cần giải quyết với DR-submodular khi: Cần đặt các cảm biến để thu thập thông tin, ta lựa chọn đặt nhiều cảm biến có năng lượng tiêu thụ thấp để tối thiểu năng lượng tiêu hao thay vì đặt một cảm biến có thể thu được thông tin tốt hơn nhưng tốn năng lượng hơn. Tương tự khi phân bổ tài nguyên, ngân sách, người phân phối sẽ có xu hướng chọn nhiều lần những người dùng hoặc đại lý tiềm năng nào đó cho lợi nhuận tốt.

Các tình huống tương tự như trên thúc đẩy các nhà nghiên cứu tìm hiểu các phiên bản tổng quát hóa của tính submodular và lợi nhuận hiệu suất giảm dần được định nghĩa trên các *lưới nguyên (Integer lattice)*, trở thành DR-submodular. Nếu ta xét thêm điều kiện giới hạn về lực lượng ít nhất mà vẫn thu được tối thiểu lượng thông tin α, bài toán trở thành thể hiện của bài toán DRSC. Việc nghiên cứu giải bài toán Phủ Submodular đơn điệu trên lưới nguyên là một bài toán mới hiện nay, và có nhiều thách thức đặt ra:

- Mặc dù đã có nhiều công trình nghiên cứu để giải SC, thách thức khi giải DRSC gặp phải là các tính chất của submodular sẽ không còn giữ nguyên khi xét trên lưới nguyên mà có sự mở rộng và khái quát hơn.
- Bài toán SC là bài toán NP-khó, cho nên bài toán DRSC cũng là bài toán NP-khó;
- Vì không gian tìm kiếm là không gian nhiều chiều, $\mathbb{Z}_+^V$, nên số tổ hợp cần tìm sẽ tăng lên rất nhiều, ảnh hưởng tới thời gian tìm kiếm lời giải của thuật toán.
- Đây là hướng nghiên cứu mới, cần đóng góp các thuật toán xấp xỉ có giá trị về mặt lý thuyết.

Từ các thách thức này, luận án đặt vấn đề nghiên cứu một hướng tiếp cận giải bài toán tối ưu hàm submodular phổ biến hiện nay đó là thiết kế thuật toán song song nhằm giảm thời gian tính toán. Đồng thời, luận án đề xuất *thuật toán xấp xỉ tiêu chí kép (bi-criteria approximation algorithm)* hiệu quả, cải tiến hơn so với các tác giả sử dụng giải pháp tương tự cho hai bài toán DRSC và SC.

4.2. Thuật toán xấp xỉ cho bài toán DRSC

4.2.1. Kết quả mới của luận án

Việc tìm được lời giải tối ưu sao cho chi phí nhỏ nhất với ràng buộc hàm $f(\cdot)$ là hàm submodular làm cho bài toán trở nên khó khăn hơn khi phải thỏa mãn cả hàm chi phí và hàm submodular. Do đó, luận án sử

dụng thuật toán xấp xỉ tiêu chí kép để tìm một thuật toán xấp xỉ hiệu quả giải bài toán DRSC. Thuật toán xấp xỉ tiêu chí kép được đưa ra khi nới lỏng các ràng buộc của bài toán, cho lời giải có hai tỉ lệ xấp xỉ, được định nghĩa như sau:

Định nghĩa 4.3 (Thuật toán xấp xỉ tiêu chí kép (σ_1, σ_2))**.** Thuật toán là xấp xỉ tiêu chí kép với tỉ lệ (σ_1, σ_2) cho bài toán DRSC khi nó trả lại lời giải $\mathbf{x}$ thỏa mãn $\|\mathbf{x}\|_1 \leq \sigma_1 \cdot \|\mathbf{o}\|_1$ và $f(\mathbf{x}) \geq \sigma_2 \cdot \alpha$ với $\sigma_1 > 1, \sigma_2 > 0$, và $\mathbf{o}$ là lời giải tối ưu.

Ngoài ra, thuật toán được thiết kế song song dựa trên khái niệm độ phức tạp song song dưới đây:

Định nghĩa 4.4 (Độ phức tạp song song)**.** Cho một cách ước lượng hàm f, độ phức tạp song song của một thuật toán là số vòng lặp tuần tự tối thiểu cần dùng để trong mỗi vòng lặp đó thuật toán tạo ra một số lượng đa thức các truy vấn tới ước lượng hàm f một cách độc lập với nhau.

Các đóng góp của luận án:

1. ***Về chất lượng lời giải.*** Thuật toán cho lời giải xấp xỉ với tỉ lệ hằng số là $(1+\epsilon)(1+\log(1/\lambda))$ và giá trị hàm f gần tùy ý với α (trong giới hạn sai số $\lambda > 0$). Như vậy, thuật toán của tác giả cho tỉ lệ xấp xỉ tốt hơn so với thuật toán tất định tốt nhất được đề xuất bởi Soma và Yoshida. Thuật toán của họ thực tế phụ thuộc vào $O(\log d)$, với $d = \max_{e \in E} f(\chi_e)$). Thuật toán của tác giả cũng đảm bảo xấp xỉ tốt hơn thuật toán ngẫu nhiên tốt nhất hiện nay đề xuất bởi Ran và cộng sự và phép dẫn về của Ene và Huy L. Nguyen. Tỉ lệ xấp xỉ của thuật toán của họ phụ thuộc vào $O(H(\min\{\max_{e \in S} f(e), \alpha\}))$. Ngoài ra, lưu ý rằng thuật toán của họ chỉ giải quyết trường hợp hàm mục tiêu mang giá trị nguyên.

2. ***Về độ phức tạp.*** Thuật toán ngẫu nhiên cho độ phức tạp song song tốt nhất là thuật toán kết hợp giữa công trình của Ran và cộng sự vớicủa Ene và Huy L. Nguyen. So với các kết hợp này, thuật toán trong luận án cho độ phức tạp song song thấp hơn 1 hệ số là $\Omega(\log(m)\log^2(mn\log B)(1+\log(\log B)/\log(n)))$, cho độ phức tạp truy vấn thấp hơn 1 hệ số là $\Omega(\min\{\log(n)\log(B)/\log(nB), n\} \cdot \log(m)\log^2(mn\log(B)))$ với $m = f(B \cdot \mathbf{1})$. So sánh với thuật toán tất định cho độ phức tạp truy vấn tốt nhất của Soma, thuật toán cho độ phức tạp song song và độ phức tạp truy vấn ít hơn 1 hệ số lần lượt là $\Omega(\min\{n/\log(n), \log(B)\})$ và $\Omega(n\log(B)(1+\log(B)))$.

3. ***Về giải bài toán SC.*** Để giải bài toán SC, thuật toán cho lời giải xấp xỉ tiêu chí kép $((1+\epsilon)(1+\log 1/\lambda), 1-\lambda)$ với độ phức tạp song song là $O(\log n)$ và độ phức tạp truy vấn là $O(n\log n)$. Đây là các đảm bảo lý thuyết có giá trị và vượt trội hơn so với các thuật toán xấp xỉ hiện nay.

4.2.2. Thuật toán chính: BA

Luận án xây dựng thuật toán xấp xỉ tiêu chí kép **BA** (*Bi-criteria Algorithm*) là thuật toán chính dựa trên thuật toán **AdaptDRSC**. Thuật toán **AdaptDRSC** nhận hai tham số đầu vào cho trước $\epsilon \in (0,1), \lambda \in (0,1)$ và một giá trị tiên đoán của giá trị tối ưu, v, sao cho $(1-5\epsilon_1)v \leq \mathsf{opt} \leq v$, với $\epsilon_1 = \epsilon/50$ và trả về một lời giải dự tuyển tương ứng.

Định lý 4.1. *Với $\lambda \in (0,1)$, $\epsilon \in (0,1)$, thuật toán* **BA** *chạy trong $O\left(\log(1/\lambda)\log(n)/\epsilon^2\right)$ vòng lặp tuần tự. Nó tốn $O\left((n+\log(n)\log(B)\log(1/\epsilon)/\epsilon^3)\log(nB)\log(1/\lambda)/\epsilon^3\right)$ truy vấn và trả về lời giải xấp xỉ tiêu chí kép đạt tỉ lệ $((1+\epsilon)(1+\log(1/\lambda)), 1-\lambda)$ theo kỳ vọng.*

Đối với bài toán SC, tức là $B = 1$, thuật toán này vẫn đảm bảo cùng tỉ lệ xấp xỉ tiêu chí kép và độ phức tạp song song. Độ phức tạp truy vấn lúc này giảm xuống còn $O\left(n\log(n)\log(1/\lambda)/\epsilon^3\right)$. Từ Định lý 4.1, có hệ quả cho SC dưới đây.

Algorithm 5 : $\mathbf{BA}$

Input: $f : \mathbb{Z}_+^V \mapsto \mathbb{R}_+$, a positive integer B, α, $\epsilon \in (0,1), \lambda \in (0,1)$
Output: vector $\mathbf{x}$

$\epsilon_1 \leftarrow \frac{\epsilon}{50}, C \leftarrow \{\frac{1}{(1-5\epsilon_1)^j} : 0 \leq j \leq \log_{\frac{1}{1-5\epsilon_1}}(nB)\}$
for all $v \in C$ **(in parallel) do**
  $\mathbf{x}^v \leftarrow \mathbf{AdaptDRSC}(f, B, \alpha, \epsilon_1, \lambda, v)$
end for
$\mathbf{x} \leftarrow \arg\min_{v \in V : f(\mathbf{x}^v) \geq (1-\lambda)\alpha} \|\mathbf{x}^v\|$
return $\mathbf{x}$

Hệ quả 4.1. *Đối với trường hợp của bài toán* SC*, tức là* $B = 1$*, thuật toán* $\mathbf{BA}$ *có độ phức tạp song song là* $O\left(\log(n)\log(1/\lambda)/\epsilon^2\right)$*, cần độ phức tạp của truy vấn là* $O\left(n\log(n)\log(1/\lambda)/\epsilon^3\right)$ *và trả về lời giải xấp xỉ tiêu chí kép với tỉ lệ* $((1+\epsilon)(1+\log(1/\lambda)), 1-\lambda)$ *theo kỳ vọng.*

KẾT LUẬN

Luận án nghiên cứu thuật toán xấp xỉ cho bài toán tối ưu tổ hợp với dữ liệu lớn, trong đó tập trung nghiên cứu một số bài toán thường gặp trong phân tích dữ liệu. Từ đó, khái quát thành bài toán tối đa hàm submodular có ràng buộc và các bài toán mở rộng khác. Cụ thể, các bài toán mà luận án đã trình bày gồm: Bài toán tối đa hàm k-submodular với ràng buộc chi phí - kSMK; Bài toán tối đa hàm submodular với ràng buộc chi phí có nhiễu - SMKN; Bài toán Phủ Submoudlar trên lưới nguyên - DRSC. Với mỗi bài toán, luận án đưa ra các thuật toán xấp xỉ cho tỉ lệ xấp xỉ cạnh tranh, giảm độ phức tạp truy vấn hoặc độ phức tạp song song, qua đó góp phần giảm thời gian chạy. Các đóng góp của luận án bao gồm:

1. Đối với bài toán kSMK, luận án đề xuất thuật toán $\mathbf{FA}$ đưa độ phức tạp truy vấn về tuyến tính, trên cơ sở đó đề xuất thuật toán cải tiến $\mathbf{IFA}$, cải tiến tăng cường $\mathbf{IFA+}$ giải quyết bài toán với trường hợp hàm mục tiêu đơn điệu. $\mathbf{IFA+}$ cho tỉ lệ xấp xỉ tốt hơn so với thuật toán tốt nhất hiện nay, với độ phức tạp truy vấn giảm xuống một hệ số. Luận án cũng mở rộng kết quả nghiên cứu giải quyết với hàm mục tiêu không đơn điệu, đề xuất thuật toán xấp xỉ tuyến tính tăng cường $\mathbf{RLA}$, cho tỉ lệ xấp xỉ tốt tương đương với thuật toán tốt nhất hiện nay nhưng độ phức tạp truy vấn cũng giảm xuống một hệ số.

2. Đối với bài toán SMKN, luận án đề xuất thuật toán tham lam $\mathbf{GUN}$ giải quyết bài toán. Để cải thiện tốc độ của thuật toán và không gian lưu trữ, luận án đã đề xuất thuật toán luồng $\mathbf{NS}$ để giải bài toán này.

3. Đối với bài toán DRSC, luận án xây dựng một thuật toán song song tiêu chí kép $\mathbf{BA}$ cho độ phức tạp truy vấn và độ phức tạp song song tốt hơn hẳn các thuật toán tốt nhất hiện nay. Các kết quả có thể áp dụng được sang bài toán SC và cho kết quả vượt trội hơn các kết quả hiện nay.

Trong tương lai, NCS tiếp tục mở rộng nghiên cứu các bài toán biến thể khác của bài toán tối đa hàm submodular. Các vấn đề có thể mở rộng nghiên cứu: Tối đa hàm submodular với ràng buộc d-knap; Tối đa hàm non-submodular; Bài toán kSMK trong môi trường nhiễu; Các bài toán SM giải trên lưới nguyên. Các hướng nghiên cứu bao gồm: Tiếp tục phát triển các thuật toán hiệu quả cho các bài toán; Nghiên cứu các ứng dụng trong học máy và trí tuệ nhân tạo là thể hiện của các bài toán tối đa hàm submodular.

DANH MỤC CÔNG TRÌNH KHOA HỌC CỦA TÁC GIẢ LIÊN QUAN ĐẾN LUẬN ÁN

1. Canh V Pham, **Dung KT Ha**, Huan X Hoang and Tan D Tran, *Fast Streaming Algorithms for k-Submodular Maximization under a Knapsack Constraint*, IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA), 2022 (**ISI/RANK A**);

2. **Dung T.K. Ha**, Canh V. Pham, Tan D. Tran, *Improved Approximation Algorithms for k-Submodular Maximization under a Knapsack Constraint*, Computers & Operations Research (**ISI/Q1**);

3. **Dung T. K Ha**, Canh V. Pham, Tan D. Tran and Huan X. Hoang. *Robust Approximation Algorithms for Non-monotone k-Submodular Maximization under a Knapsack Constraint*, The 15th IEEE International Conference on KNOWLEDGE AND SYSTEMS ENGINEERING (KSE 2023);

4. **Dung T. K Ha**, Canh V. Pham and Huan X. Hoang, *Submodular Maximization Subject to a Knapsack Constraint Under Noise Models*, Asia-Pacific Journal of Operational Research, Tập 39, Số 6, 2022 (**ISI/Q3**);

5. Canh V. Pham and **Dung T.K. Ha**. *A note for approximating the submodular cover problem over integer lattice with low adaptive and query complexities*. Information Processing Letters, Volume 182, 2023, (**ISI/Q3**).