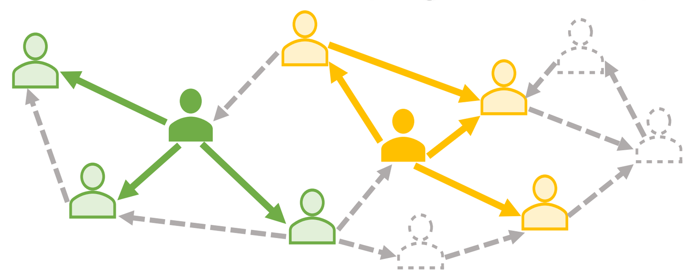
Efficient and Effective Algorithms for A Family of Influence Maximization Problems with A Matroid Constraint

Yiqian Huang, Shiqi Zhang, Laks V.S. Lakshmanan, Wenqing Lin, Xiaokui Xiao, Bo Tang

Problem Formulation

Influence Maximization (IM): in a social network, find a set of k users S that maximizes the expected number of *influenced* users: $\sigma(S)$, where σ is defined via the mechanism called the diffusion model (e.g., IC or LT).

Example of IM:
$S = \{\ \}$; $k = 2$;
: (excepted) influenced users



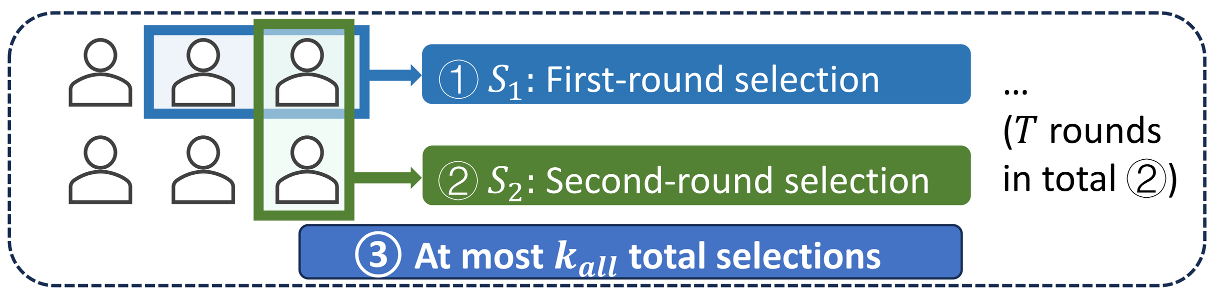
Matroid: it models constraints across multiple selected sets (see below).

Formal definition: a matroid is a pair $(U, \mathbb{I})$, where U is the ground set and $\mathbb{I}$ is a collection of subset of U (which is all feasible solutions) satisfying three axioms: (i) $\mathbb{I} \neq \emptyset$; (ii) $I \in \mathbb{I}, J \subseteq I \Rightarrow J \in \mathbb{I}$; (iii) $I, J \in \mathbb{I}, |J| < |I| \Rightarrow \exists x \in I \backslash J$ such that $J \cup \{x\} \in \mathbb{I}$.

We study **IM subject to a *general matroid* constraint (IM-GM):**

Maximize $\sigma(S)$ subject to $S \subseteq U, S \in \mathbb{I}$.

Examples of IM-GM with matroid $(U, \mathbb{I})$:



Example 1 (IM (only ①; $T = 1, k_{all} = k$)) U: all users; $\mathbb{I}$: $|S| \leq k$ (at most k users) [uniform matroid]

Example 2 (Multi-round (only ①②; $k_{all} = +\infty$)) U: all users × T rounds; $\mathbb{I}$: $\forall i, |S_i| \leq k_i$ (at most k_i users in round i) [partition matroid]

Example 3 (Multi-round + an overall budget (①②③)) U: all users × T rounds; $\mathbb{I}$: $\forall i, |S_i| \leq k_i$ *and* $\sum_{i=1}^{T} |S_i| \leq k_{all}$ [general matroid]

Previous Work

1. IM-GM is an instance of monotone submodular maximization under a matroid constraint (thus NP-Hard).
2. Existing $(1 - 1/e - \epsilon)$-approximation algorithms adopt hill-climbing on the following *multilinear extension* of $\sigma(\cdot)$:

$$F([x_1 \ \dots \ x_{|U|}]) = \sum_{S \subseteq U} \prod_{i \in S} x_i \prod_{j \notin S} (1 - x_j) \cdot \sigma(S).$$

- Prior work (Calinescu et al. SIAM J. Comput.'11; Badanidiyuru et al. SODA'14) estimates $F([x_1 \ \dots \ x_{|U|}])$ by sampling, as it involves all $2^{|U|}$ possible S.

Main Idea

1. Approximating σ: Adapt the previous approach (RR sets) to approximate $\sigma(S)$ via set cover; specifically, we generate a collection $\mathcal{R}$ of RR sets, then

$$\sigma(S) \approx C \cdot \sum_{R \in \mathcal{R}} \mathbf{1}\{S \cap R \neq \emptyset\}. \quad (C = \tfrac{|U|}{|\mathcal{R}|} \textit{ is constant})$$

2. Evaluating F: Hill-climbing on $F(\cdot)$, with **fast & exact assessment of its value**, motivated by plugging the set cover into $F(\cdot)$:

$$F([x_1 \ \dots \ x_{|U|}]) \approx C \cdot \sum_{R \in \mathcal{R}} \underbrace{\sum_{S \subseteq U} \prod_{i \in S} x_i \prod_{j \notin S} (1 - x_j)\, \mathbf{1}\{S \cap R \neq \emptyset\}}_{\spadesuit}.$$

Observation: $\spadesuit = (1 - \prod_{i \in R}(1 - x_i))$ (a computable form).

Algorithms & Guarantees

1. AMP (main algorithm): Given RR sets (thus any $\Lambda_{\mathcal{R}}(S) = C \cdot \sum_{R \in \mathcal{R}} \mathbf{1}\{S \cap R \neq \emptyset\}$), AMP maximizes $\Lambda_{\mathcal{R}}(S)$ via two steps:

(i) it uses hill-climbing to maximize $F_{\Lambda_{\mathcal{R}}}$ and returns a fractional solution $\vec{x} = [x_1 \ \dots \ x_{|U|}]$ satisfying: (a) $\vec{x}$ is a convex combination of $\{\mathbf{1}_I : I \in \mathbb{I}\}$; (b) $x_1 + \cdots + x_{|U|}$ is maximized.

(ii) It uses a *deterministic* rounding scheme to convert $\vec{x}$ to a discrete set S satisfying: (a) $S \in \mathbb{I}$; (b) $|S|$ is maximized; (c) $\Lambda_{\mathcal{R}}(S) \geq F_{\Lambda_{\mathcal{R}}}(\vec{x})$.

2. RAMP (scalable implementation): RAMP improves AMP's efficiency by determining the number of RR sets needed to achieve a $(1 - 1/e - \epsilon)$-approximation, adapting the doubling strategy from OPIM-C (Tang et al. SIGMOD'18).

Guarantees using multi-Round IM as an example (n users, m edges; T rounds in total, k users per round):

Known that approximating $\sigma(\cdot)$ needs $L = \sum_R |R|$ time in total (Tang et al. *SIGMOD'14*);

Results of AMP:

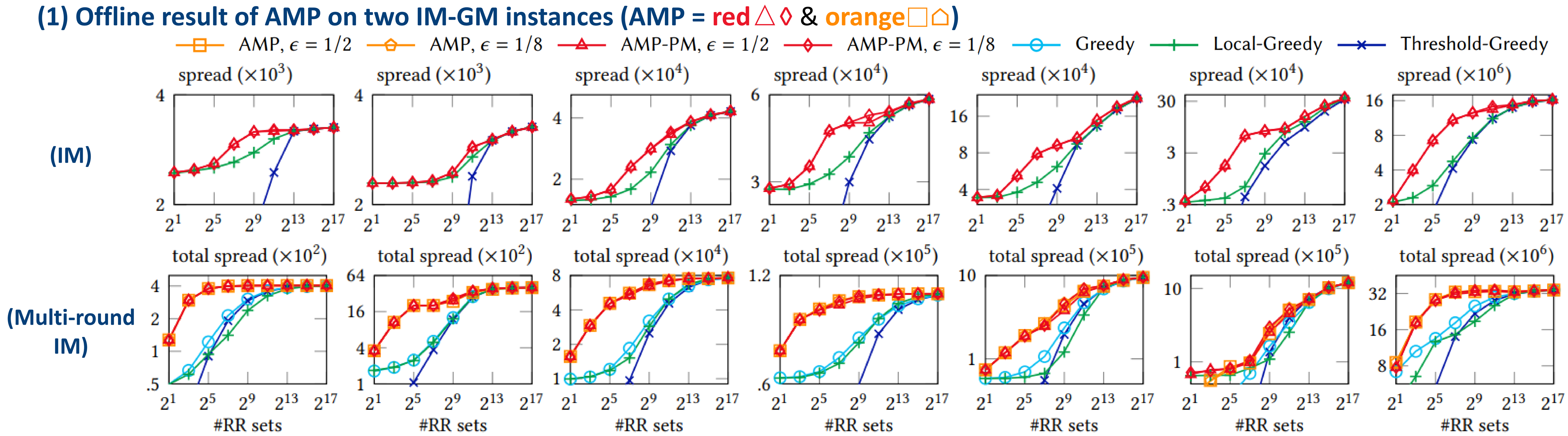
Algorithm	Time Complexity	Approximation
Calinescu et al. (*SIAM J. Comput.'11*)	$O(n^7 \cdot L)$	$1 - 1/e - \epsilon$
Badanidiyuru et al. (*SODA'14*)	$O(nkT \cdot \epsilon^{-4} \cdot L)$	
Our algorithm **(AMP)**	$O(k \cdot \epsilon^{-1} \cdot L)$	

Results of RAMP (to achieve $(1 - 1/e - \epsilon)$-approximation w.p. $1 - \delta$):

Time complexity = $O\left(T^2 k \epsilon^{-2} (\ln n + \ln(1/\delta)) \cdot (m + k\epsilon^{-1} n)\right)$

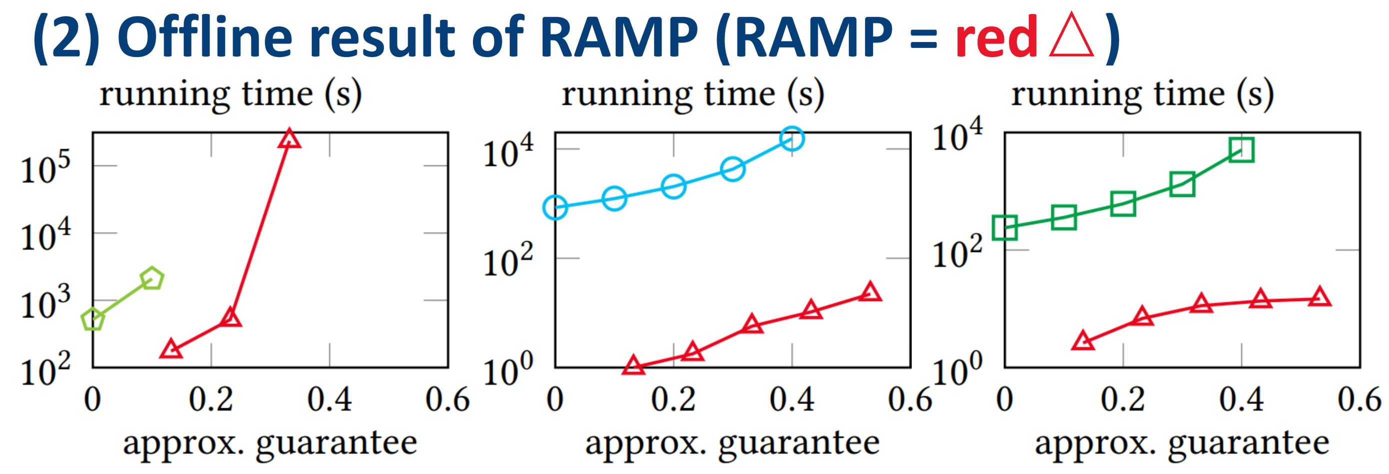
Representative Experimental Results

(1) Offline result of AMP on two IM-GM instances (AMP = red △◊ & orange □⌂)



(a) Facebook (n=4.0K m=176.4K)
(b) Wiki (n=7.1K m=103.6K)
(c) Twitter (n=81.3K m=1.7M)
(d) Google+ (n=107.6K m=13.7M)
(e) Pokec (n=1.6M m=30.6M)
(f) LiveJournal (n=4.8M m=69.0M)
(g) Twitter-large (n=41.7M m=1.5B)

(2) Offline result of RAMP (RAMP = red △)



(a) RM (Twitter-large)
(b) MRIM (Wiki)
(c) AdvIM (Wiki)

(3) Online deployment for a Tencent game

Task: recommend in-game products (maps) to each user
Scale: 0.7 million users and 400 maps
Metric: Increase in engaged user–map pairs (vs. control group, $K = 10^3$)
Result:

Algorithm	RM-A	RAMP (ours)
Δ of pairs	20K	**160K**

NUS National University of Singapore · PyroWis · THE UNIVERSITY OF BRITISH COLUMBIA · Tencent · 南方科技大学 SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY