Swarm and Evolutionary Computation 106 (2026) 102428

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Fairness-constrained influence maximisation via multi-objective optimisation

Ziying Zhao [a], Weihua Li [a,*], Jing Ma [a], Jianhua Jiang [b], Quan Bai [c], Wen Gu [d]

[a] Department of Data Science and Artificial Intelligence, School of Engineering, Computer and Mathematical Sciences, Auckland University of Technology, 55 Wellesley Street East, Auckland Central, Auckland, 1010, New Zealand
[b] Jilin Province Key Laboratory of Fintech, Jilin University of Finance and Economics, 3699 Jingyue Street, Changchun, 130117, Jilin Province, PR China
[c] The School of Information and Communication Technology, University of Tasmania, Churchill Avenue, Sandy Bay, Hobart, 7005, Tasmania, Australia
[d] Department of Computer Science and Engineering, Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya, 466-8555, Aichi, Japan

ARTICLE INFO

Keywords:
FairWolf
Fair influence maximisation
Grey wolf optimiser
Multi-objective optimisation
Social networks

ABSTRACT

The Influence Maximisation (IM) problem seeks to select a set of seed nodes to maximise information diffusion in a network. While existing approaches have achieved significant improvements in overall diffusion, they often overlook fairness across communities, which can result in biased dissemination and the exclusion of disadvantaged groups. To address this, we define the Fair Multi-objective Influence Maximisation (FMOIM) problem, which jointly optimises influence spread and equity fairness. Equity Fairness is modelled at the community level as the alignment between the realised diffusion-benefit distribution and a desired reference allocation. Jensen–Shannon divergence (JSD) similarity quantifies distributional deviation from the reference allocation, while Jain's fairness index characterises the evenness of benefit allocation across communities. To solve FMOIM, FairWolf is proposed as a problem-driven discrete multi-objective optimisation model for fairness-aware influence maximisation. It reformulates the Grey Wolf Optimiser dynamics to search directly over fixed-budget seed sets under community-level fairness objectives, capturing the spread and fairness trade-off. FairWolf incorporates three components: (i) a discrete position-updating mechanism tailored to seed-set construction, (ii) an Explorer-Augmented Leader Selection strategy that enhances population diversity while maintaining convergence pressure, and (iii) a Hypervolume (HV)-triggered perturbation mechanism that adaptively mitigates stagnation in non-convex multi-objective search spaces. Experiments on eight real-world networks demonstrate that the FairWolf model consistently outperforms state-of-the-art baselines, yielding a higher HV value and more uniformly distributed Pareto fronts. These results demonstrate its effectiveness and practicality for fairness-aware diffusion in applications such as viral marketing, public health, and resource allocation.

1. Introduction

The growing role of social networks has transformed the way information spreads across society. Platforms such as X,[1] Facebook,[2] and WeChat[3] enable information to reach millions of users almost instantaneously, reshaping the dynamics of communication and interaction. Influence Maximisation (IM), which focuses on selecting a set of seed nodes to maximise information diffusion, has therefore become a fundamental research problem with broad applications in domains such as viral marketing [1], public health campaigns [2], political communication [3], and social recommendation systems [4]. By identifying and optimising influential nodes, organisations and policymakers can achieve more effective outreach, resource allocation, and strategic decision-making.

The IM problem was formally introduced by Kempe et al. [5] in the context of diffusion models such as the Independent Cascade (IC) and Linear Threshold (LT) models, revealing that the influence function is submodular and thus admits approximation guarantees through greedy selection. While subsequent studies developed several classical approaches such as CELF [6], degree-based heuristics [7], and

* Corresponding author.
E-mail addresses: ziying.zhao@autuni.ac.nz (Z. Zhao), weihua.li@aut.ac.nz (W. Li), jing.ma@aut.ac.nz (J. Ma), jjh@jlufe.edu.cn (J. Jiang), quan.bai@utas.edu.au (Q. Bai), wgu@nitech.ac.jp (W. Gu).

[1] https://x.com/.
[2] https://www.facebook.com/.
[3] https://www.wechat.com/.

https://doi.org/10.1016/j.swevo.2026.102428
Received 12 October 2025; Received in revised form 7 May 2026; Accepted 26 May 2026
Available online 2 June 2026

centrality-based methods (e.g., betweenness [8]), most of them mainly focused on maximising the overall number of influenced nodes, without considering how influence is distributed across different groups in the network.

In reality, social networks are often organised into communities, in which nodes within the same community are more densely connected than those outside it. These structures significantly influence the diffusion process: communities with stronger internal ties tend to acquire information earlier and more comprehensively, whereas communities with weaker connections may receive it only partially or at later stages. Such disparities are particularly problematic in applications where fairness is critical. For example, in health awareness campaigns, disadvantaged groups must have timely and sufficient access to life-saving information [2]; and in digital advertising, discriminatory targeting has raised regulatory and ethical concerns [9]. These issues have motivated the emergence of **Fair Influence Maximisation (FIM)**, which extends the IM framework by explicitly incorporating fairness objectives into the diffusion process.

Several fairness notions have been introduced in the literature, including Early-adopter Fairness [10], Maximin Fairness [11], Equity Fairness [12], Welfare Fairness [13], and time-critical fairness [9]. These notions capture different fairness goals, but each has limitations when applied to community-structured diffusion. For instance, early-adopter or time-critical criteria emphasise when communities receive information, yet do not directly characterise how diffusion benefits are allocated across communities over the whole process. Maximin-style objectives protect the worst-off group but can be overly conservative and may lead to large losses in overall spread. Among these, Equity Fairness proposed by Farnadi et al. [12] has received particular attention. It requires that the expected influence spread within each community should be proportional to its population size. However, this proportionality assumption overlooks differences in network connectivity and diffusion dynamics, so some communities may still be over- or underrepresented, even when proportionality holds only at the aggregate level. Moreover, relying on a single fairness criterion often offers limited interpretability. It may be unclear whether unfairness arises from misalignment with the proportional target at the distribution level or from systematic over- or under-allocation to specific communities.

To handle these issues, we revisit equity fairness at the community level and operationalise it as proportional allocation of diffusion benefits across communities. We then adopt two complementary and interpretable measures. Jensen–Shannon divergence (JSD) similarity [14] quantifies the distributional alignment between the realised community benefit-share distribution and the population-share reference implied by community sizes. It indicates the extent to which the overall allocation deviates from the intended proportional target. Jain's fairness index [15] captures community-wise proportional balance by assessing whether communities receive diffusion benefits roughly proportional to their population shares. In other words, it examines whether the benefit-to-population ratios are consistent across communities. This helps reveal systematic over- or under-allocation to particular communities. The two measures are complementary. Global distributional alignment alone may still conceal community-level disproportionality, while proportional balance alone may still permit an overall mismatch between the realised allocation and the population reference. Together, they provide a more informative and actionable fairness assessment, helping diagnose whether particular communities are persistently underserved relative to proportional allocation and guiding fairness-aware seed selection. Based on this, we formally define the **Fair Multi-objective Influence Maximisation (FMOIM)** problem, which jointly optimises overall influence spread and equity fairness.

To address the FMOIM problem, we propose and develop a novel model, called **FairWolf**, which is a problem-driven discrete multi-objective optimisation model for fairness-aware influence maximisation. The key design is to reformulate the Grey Wolf Optimiser dynamics to search directly over fixed-budget seed sets under community-level fairness objectives, thereby capturing the spread-fairness trade-off in the seed-selection space. FairWolf is designed around the discrete nature of seed-set construction and the fairness-aware objective structure, and the search naturally produces diverse Pareto-optimal seed sets with meaningful fairness interpretation. FairWolf incorporates three components: (i) a discrete position update that preserves the seed-set budget by construction, (ii) an Explorer-Augmented Leader Selection mechanism that sustains diversity while retaining convergence pressure, and (iii) an HV-triggered perturbation mechanism that adaptively mitigates stagnation in non-convex multi-objective landscapes.

The main contributions of this paper are summarised as follows:

- We revisit equity fairness and introduce two complementary and interpretable measures (JSD similarity and Jain's fairness index) to characterise community-level proportional allocation in influence diffusion.
- We formalise the Fair Multi-objective Influence Maximisation (FMOIM) problem that jointly optimises influence spread and equity fairness, making the spread-fairness trade-off explicit.
- We propose FairWolf as a problem-driven discrete multi-objective optimisation model for fairness-aware seed selection, and conduct extensive experiments on real-world networks to demonstrate its effectiveness.

The remainder of this paper is organised as follows. Section 2 reviews the literature on fair influence maximisation. Section 3 provides the problem description and essential definitions. Section 4 formalises the Fair Multi-objective Influence Maximisation (FMOIM) problem. Section 5 details the proposed FairWolf model. Section 6 reports experimental results and provides a comprehensive discussion. Finally, Section 7 concludes the paper and outlines future research directions.

2. Related work

Research on influence maximisation has been extensively developed since the seminal work of Kempe et al. [5]. Early solutions relied on greedy selection under submodular influence functions, which provides approximation guarantees but typically requires repeated diffusion simulations and is therefore computationally demanding [6]. To improve efficiency, heuristic methods such as degree-based strategies [7], centrality-based heuristics (e.g., betweenness [8]), and reverse influence sampling (RIS) [16] have been developed, substantially reducing computational overhead while retaining competitive accuracy.

Beyond greedy and heuristic pipelines, metaheuristic and learning-based approaches have been explored for their flexibility in handling combinatorial search and practical constraints. Tang et al. [17] proposed a discrete shuffled frog-leaping algorithm for seed selection. Salavati et al. [18] applied ant colony optimisation for influential node detection. Zhang et al. [19] developed a constrained evolutionary algorithm with influence indicators to better handle budget restrictions. Ma et al. [20] introduced evolutionary deep reinforcement learning by combining Q-learning with evolutionary optimisation, and Tian et al. [21] explored deep reinforcement learning for topic-aware IM. Overall, these methods demonstrate the flexibility of population-based and learning-based search for IM, particularly when the decision space is discrete and diffusion evaluation is costly.

While the above advances improve diffusion efficiency, they typically optimise total spread and do not explicitly control how diffusion benefits are distributed across groups. This limitation motivates fair influence maximisation (FIM), which incorporates fairness considerations into seed selection and diffusion outcomes. Existing FIM notions cover multiple perspectives. Early-adopter and outreach fairness focus on balanced exposure across groups in seeding and diffusion outcomes [10]. Time-critical fairness further accounts for diffusion deadlines, showing that time constraints can exacerbate disparities, and proposes fairness-aware formulations subject to budget and coverage requirements [9].

Maximin fairness aims to protect the least-advantaged community, often via multi-objective submodular optimisation and related constraints [11]. Welfare-oriented formulations introduce inequality aversion to tune the trade-off between fairness and total influence [13], and individual-level fairness objectives have also been explored (e.g., ϕ-mean fairness) [22]. Equity fairness targets proportional allocation, requiring that the expected influence received by each community is proportional to its population size [23]. Farnadi et al. [12] further proposed a unifying framework that connects maximin, equality, equity, and diversity within a single optimisation view.

However, these formulations often emphasise a single aspect of fairness, which can limit interpretability in heterogeneous networks. Structural diffusion advantages may yield systematic over- or under-allocation across communities even when aggregate targets appear satisfied. These observations motivate fairness objectives that are both community-aware and easier to interpret in terms of allocation behaviour.

Building on fairness formulations, recent work has developed algorithmic frameworks for FIM, especially in multi-objective and community-aware settings. Razaghi et al. [24] proposed SetMOGWO, a discrete multi-objective grey wolf optimiser that jointly optimises influence spread together with fairness-related objectives, and assesses trade-offs using the Price of Fairness (POF) metric [11,25]. Ma et al. [26] introduced a community-based evolutionary algorithm that explicitly leverages community structure to improve maximin-oriented fairness. Zhao et al. [27] extended the grey wolf framework to optimise overall influence and group-fair objectives based on initial activation proportion differences (IAPD). In addition, mixed-integer programming has been used to solve unified fairness formulations when exact optimisation is feasible, trading efficiency for exactness [12].

Despite these advances, current research on FIM remains limited. First, fairness evaluation in diffusion is often hard to interpret at the community level, especially under heterogeneous connectivity and diffusion dynamics. Second, multi-objective FIM solvers can remain computationally costly because diffusion-based evaluation dominates runtime. This makes it important to design optimisation models that search effectively over fixed-budget seed sets under fairness objectives. These challenges highlight the need for fairness metrics that better characterise community-level allocation behaviour in diffusion, as well as optimisation methods that can address such objectives effectively.

3. Preliminaries

In this section, we introduce the problem description and the essential definitions required for subsequent analysis.

3.1. Problem description

This paper addresses the Fair Multi-Objective Influence Maximisation (FMOIM) problem, which aims to identify a set of Pareto-optimal solutions that promotes the equitable dissemination of influence across a network. Each solution corresponds to a node set $S^{(i)} \subseteq V$ of fixed size $|S^{(i)}| = k$. The optimiser explores multiple such candidates, and the collection of all generated solutions is denoted by $\mathcal{S}$. From $\mathcal{S}$, we extract the Pareto front $PF \subseteq \mathcal{S}$, which constitutes the final output of the optimisation procedure. The Pareto front represents the set of trade-off optimal solutions balancing influence spread and fairness.

We consider an undirected social network $G = (V, E)$, where V denotes the set of nodes and E represents the set of all edges. An edge $(v_i, v_j) \in E$ represents the potential diffusion of influence between nodes v_i and v_j. The nodes in network G are partitioned into distinct communities, denoted as $C_j \subseteq V$, and each node belongs to exactly one community.

Let $S^{(i)}$ denote the candidate node set of budget k, which serves as the starting point for influence diffusion under the Independent Cascade (IC) Model [5]. In this model, each activated node has exactly one opportunity to activate each of its inactive neighbours with a specified propagation probability p. The IC model is adopted because it is a canonical stochastic diffusion model in influence maximisation and has been widely used to evaluate probabilistic information propagation. Following prior work [28], we approximate the overall influence spread using a 2-hop diffusion horizon. This setting provides a computationally efficient approximation during repeated population-based optimisation, where each candidate seed set must be evaluated many times. Thus, the maximum diffusion step in the IC model is fixed at $max\_step = 2$. The IC model is executed over N Monte Carlo simulations. The set of nodes activated by $S^{(i)}$ in the nth simulation ($n \in 1, 2, \ldots, N$) is denoted as $\text{activated}(S^{(i)})_n$.

Table 1
Mathematical notations and descriptions.

Category	Description
Sets	
$G = (V, E)$	Undirected social network
V	Set of all nodes in G
E	Set of all edges in G
$C_j \subseteq V$	Node set of the jth community // $j = 1, \ldots, m$ and $\bigcup_{j=1}^{m} C_j = V$
Parameters	
N	Number of Monte Carlo simulations
p	Propagation probability
$\|V\|$	Total number of nodes in the network
$\|C_j\|$	Number of nodes in the jth community
m	Total number of communities in the network
k	Node set budget (number of selected nodes)
Decision variables and solutions	
$S^{(i)} \subseteq V$	The ith candidate seed set (solution) with budget $\|S^{(i)}\| = k$
$\mathcal{S}$	The set of all candidate solutions generated by the optimiser
$PF \subseteq \mathcal{S}$	The Pareto Front // the final output of the optimisation process
Objective-related metrics	
$\text{activated}(S^{(i)})_n$	Nodes activated by $S^{(i)}$ in the nth simulation
$\text{avgAct}(C_j)$	Average number of activated nodes in community j
$P = \{P_j\}_{j=1}^{m}$	Observed proportions of activated nodes in each community
$Q = \{Q_j\}_{j=1}^{m}$	Community-size proportions in the network
M	Number of objectives in FMOIM problem
$f_1(S^{(i)})$	Influence spread objective of candidate seed set $S^{(i)}$
$f_2(S^{(i)})$	Equity fairness objective of candidate seed set $S^{(i)}$
$\Delta f_1(v \mid S^{(i)})$	Marginal contribution of candidate node v to the spread objective with respect to seed set $S^{(i)}$

The notations and parameters used throughout this paper are summarised in Table 1.

3.2. Fundamental definitions

Definition 1 (*Community Structure*)**.** Let $G = (V, E)$ denote an undirected social network. The node set V is partitioned into m disjoint subsets $\{C_1, C_2, \ldots, C_m\}$, where $C_i \cap C_j = \emptyset$ for $i \neq j$ and $\bigcup_{j=1}^{m} C_j = V$. Each C_j represents a community in the network, where nodes share common attributes or structural proximity. Every node $v \in V$ belongs to exactly one community C_j.

In this paper, communities are used as structural fairness groups for evaluating diffusion-benefit allocation. They are not assumed to be immutable ground-truth demographic categories. Rather, they provide a group-level basis for measuring whether the realised diffusion benefits are proportionally allocated across different parts of the network. The proposed formulation is therefore defined with respect to a given group partition.

Definition 2 (*Independent Cascade Model*)**.** The Independent Cascade (IC) model describes stochastic influence diffusion in a network [5].

Starting from an initial node set $S^{(i)} \subseteq V$, each newly activated node has exactly one chance to activate each of its inactive neighbours independently with probability p. Activated nodes remain active and may trigger further activations in subsequent steps. In this work, following prior studies, we restrict the diffusion horizon to two steps (max step = 2), providing a computationally efficient approximation of the overall influence spread.

Definition 3 (*Pareto Dominance*)**.** Given two solutions $x, y \in S$ with objective vectors $f(x) = (f_1(x), f_2(x), \ldots, f_M(x))$ and $f(y) = (f_1(y), f_2(y), \ldots, f_M(y))$ in an M-objective optimisation problem, solution x is said to *Pareto dominate* y (denoted as $x \prec y$) if and only if $f_i(x) \geq f_i(y)$ for all $i \in \{1, \ldots, M\}$ and $f_j(x) > f_j(y)$ for at least one index j [29,30].

Definition 4 (*Pareto Front*)**.** The *Pareto Front* is the set of all non-dominated solutions in S. Formally,

$$PF = \{x \in S \mid \nexists y \in S,\ y \prec x\}.$$

The Pareto front represents the trade-off surface among conflicting objectives, and decision-makers can select solutions from this set according to their preferences [31].

Definition 5 (*Crowding Distance*)**.** *Crowding distance* is a diversity-preserving metric used in multi-objective optimisation [32]. For a solution on the Pareto front, it is calculated as the sum of the normalised distances between its nearest neighbours along each objective axis, thereby measuring the degree of isolation of the solution in the objective space. A larger crowding distance indicates greater isolation and helps maintain a well-distributed set of solutions across the Pareto front.

4. Fair multi-objective influence maximisation problem

This paper formally defines the Fair Multi-Objective Influence Maximisation (FMOIM) problem with two objectives. The first objective is to maximise the overall influence diffusion spread. The second objective models Equity Fairness [12], requiring the realised activation share of each community to be proportional to its population share in the network. In our formulation, Equity Fairness is quantified by jointly considering the global alignment between $P = \{P_j\}_{j=1}^{m}$ (observed activation proportions) and $Q = \{Q_j\}_{j=1}^{m}$ (community-size proportions), and the community-wise proportional balance of activations.

4.1. Constraints of the model

Let $S^{(i)} \subseteq V$ denote the node set selected in a given simulation. The selection process is required to satisfy the following constraints:

$$S^{(i)} \subseteq V, \quad |S^{(i)}| = k, \quad s_p \neq s_q,\ \forall s_p, s_q \in S^{(i)},\ p \neq q. \tag{1}$$

These constraints ensure that the selected nodes are drawn from V, that their number matches the budget k, and that they are mutually distinct. Furthermore, the nodes are selected simultaneously at the initial time step ($t = 0$), and the diffusion process is restricted to a maximum of two propagation steps.

4.2. Objective 1: Maximising influence diffusion spread

The first objective aims to maximise the global influence spread across the network. This measure is expressed as the average proportion of nodes activated during the diffusion process, as defined in Eq. (2). Specifically, for a given node set $S^{(i)}$, the influence spread is computed as the mean fraction of activated nodes over N independent simulations:

$$f_1(S^{(i)}) = \frac{1}{N} \sum_{n=1}^{N} \frac{|\text{activated}(S^{(i)})_n|}{|V|}, \tag{2}$$

where $|V|$ is the total number of nodes in G, $\text{activated}(S^{(i)})_n$ is the number of nodes activated by $S^{(i)}$ in simulation n, and N denotes the number of Monte Carlo simulations.

4.3. Objective 2: Maximising equity fairness across communities

The second optimisation objective builds on the notion of Equity Fairness in influence maximisation [12], which requires the realised activation share of each community to be proportional to its population share in the network. The community partition specifies the groups over which equity fairness is evaluated. The proposed formulation is not tied to a particular community detection algorithm. For any meaningful group partition, the realised activation distribution P and the reference allocation Q can be defined accordingly, and the fairness objective evaluates whether diffusion benefits are proportionally allocated across these groups. Therefore, a change in the partition changes the fairness-evaluation context, but does not invalidate the formulation itself. To model this principle, we formulate a composite equity objective that captures fairness from two complementary perspectives: global alignment with the population structure and community-wise proportional balance. This design provides a more complete characterisation of equity fairness.

The first component is the Jensen–Shannon Divergence (JSD) similarity [14], which evaluates the global alignment between the activation proportions and the population proportions. Let $P = \{P_j\}_{j=1}^{m}$ denote the observed activation proportions across m communities, where each P_j represents the proportion of activations in community j. Similarly, let $Q = \{Q_j\}_{j=1}^{m}$ denote the corresponding population proportions, where each Q_j represents the demographic share of community j.

$$P_j = \frac{\text{avgAct}(C_j)}{\sum_{i=1}^{m} \text{avgAct}(C_i)}, \tag{3}$$

$$Q_j = \frac{|C_j|}{|V|}, \tag{4}$$

where $\text{avgAct}(C_j)$ is the average number of activated nodes in community j, C_j denotes the node set of the jth community, $|C_j|$ is the number of nodes in that community, and $|V|$ is the total number of nodes in the network.

$$\text{JS}_{\text{sim}}(P, Q) = 1 - \frac{\text{JS}(P \parallel Q)}{\log 2}, \tag{5}$$

$$\text{JS}(P \parallel Q) = \tfrac{1}{2}\,\text{KL}(P \parallel A) + \tfrac{1}{2}\,\text{KL}(Q \parallel A), \quad A = \tfrac{1}{2}(P + Q), \tag{6}$$

where the term $\text{KL}(\cdot \parallel \cdot)$ is the Kullback–Leibler divergence [33], and A is the average distribution.

A higher $\text{JS}_{\text{sim}}(P, Q)$ indicates closer alignment between activation proportions and population proportions, reflecting equity at the distribution level.

The second component of the fairness objective is Jain's fairness index [15], which evaluates community-wise proportional balance by assessing the consistency of the ratios P_j/Q_j across communities:

$$\text{Jain}(P/Q) = \frac{\left(\sum_{j=1}^{m} P_j/Q_j\right)^2}{m \cdot \sum_{j=1}^{m} (P_j/Q_j)^2}, \tag{7}$$

where P_j/Q_j compares the observed activation share P_j of community j with its expected share Q_j implied by the population. When all ratios P_j/Q_j are close to one, activations are proportionally allocated across communities. Large deviations indicate systematic over-allocation or under-allocation to particular communities. Thus, a higher $\text{Jain}(P/Q)$ implies stronger proportional equity at the community level.

The two components are complementary: $\text{JS}_{\text{sim}}(P, Q)$ captures distribution-level alignment with the population structure, while $\text{Jain}(P/Q)$ penalises community-wise disproportionate deviations. Optimising a single measure may be insufficient, as global alignment may still conceal community-level disparities, whereas proportional balance alone may still permit global mismatches between P and Q.

Accordingly, the fairness objective is formulated as:

$$f_2(S^{(i)}) = \lambda \cdot \mathrm{JS}_{\mathrm{sim}}(P, Q) + (1 - \lambda) \cdot \mathrm{Jain}(P/Q), \quad \lambda \in [0, 1], \tag{8}$$

where both terms are normalised to $[0, 1]$. Here λ controls the trade-off between global alignment and community-wise proportionality. We use $\lambda = 0.5$ as a neutral default, and evaluate other values via a sensitivity analysis in Section 6.

4.4. Objective formulation

The Fair Multi-Objective Influence Maximisation (FMOIM) problem is formulated as:

$$\max_{S^{(i)} \subseteq V,\ |S^{(i)}|=k} \quad \left(f_1(S^{(i)}),\ f_2(S^{(i)})\right) \tag{9}$$

subject to the constraints on $S^{(i)}$. This formulation ensures both high overall diffusion and a proportionally fair allocation of influence across all communities.

4.5. Rationale for the spread-equity trade-off

The FMOIM formulation is motivated by the inherent tension between diffusion efficiency and equity-oriented allocation in community-structured networks. The spread objective f_1 measures the total activation produced by a seed set $S^{(i)}$, whereas the equity objective f_2 evaluates whether the realised community-level activation distribution P is aligned with the reference allocation Q. These two objectives are not generally equivalent in heterogeneous networks.

This can be seen from the marginal effect of candidate-node selection during seed-set construction. Let $\Delta f_1(v \mid S^{(i)})$ denote the marginal contribution of a candidate node v to the spread objective when considered with respect to the current seed set $S^{(i)}$. Suppose that two communities C_a and C_b have different diffusion efficiencies, reflected by different marginal spread contributions of their candidate nodes. There exist candidate nodes $u \in C_a$ and $v \in C_b$ satisfying

$$\Delta f_1(u \mid S^{(i)}) > \Delta f_1(v \mid S^{(i)}). \tag{10}$$

If C_a is already over-represented in the realised activation distribution and C_b is under-represented, i.e.,

$$P_a > Q_a, \qquad P_b < Q_b, \tag{11}$$

then a spread-oriented update would prefer u because it yields a larger increase in total activation. However, this choice may further increase the activation share of the already over-represented community and leave the under-represented community below its reference share. Consequently, f_1 can improve while f_2 deteriorates. Conversely, improving f_2 may require selecting nodes that increase $\mathrm{avgAct}(C_b)$ and reduce the gap between P_b and Q_b, even when their marginal contribution to f_1 is smaller.

This structural mismatch is common in real social networks, where communities may differ in size, internal density, boundary connectivity, and accessibility to influential hubs. The argument does not rely on a specific community detection method; rather, it depends on whether the resulting groups exhibit heterogeneous diffusion efficiency and accessibility. When such heterogeneity exists, seed sets that maximise total influence are not necessarily those that produce proportional community-level benefit allocation. This justifies modelling FMOIM as a bi-objective optimisation problem, where f_1 captures diffusion efficiency and f_2 captures equity-oriented allocation across communities.

Table 2
Parameters of MOGWO and FairWolf.

Parameter	Description
Common parameters	
α_g	Grid inflation parameter
β_g	Leader selection pressure parameter
$nGrid$	Number of grids per dimension
γ_g	Archive density-deletion parameter
$archive\_size$	Maximum number of solutions in the archive
pop_size	Number of wolves in the population
k	Budget of the seed node set
FairWolf - specific parameters	
τ	Window size for HV stagnation detection
ϵ_{HV}	Threshold for HV improvement
$perturb$	Perturbation ratio

5. FairWolf: A problem-driven discrete multi-objective Grey Wolf Optimiser for fair influence maximisation

In this section, we present FairWolf, a problem-driven discrete multi-objective grey wolf optimiser tailored for fair multi-objective influence maximisation (FMOIM). FairWolf builds upon the MOGWO search template [34], but is redesigned for the fixed-budget seed-set optimisation with $|S^{(i)}| = k$ and the equity fairness-aware objectives in Section 4, which makes standard continuous position updates inapplicable.

We retain two core strengths of the grey wolf hunting template: (i) a lightweight structure with few control parameters and a compact search procedure; (ii) leader-guided population dynamics that provide a simple balance between exploration and exploitation. However, when applied to FMOIM, the standard MOGWO template still faces three practical limitations in the discrete seed-set domain, including: (i) continuous-style updates that do not naturally respect the fixed-cardinality constraint $|S^{(i)}| = k$, leading to ineffective or redundant moves; (ii) archive-based leader selection that can become overly exploitative, causing leaders to concentrate in a narrow region of the objective space and reducing archive coverage; and (iii) stagnation with negligible improvement in archive quality over successive iterations.

To address these challenges, FairWolf incorporates three tailored mechanisms: (i) a discrete position updating operator that performs set-based modifications and preserves feasibility by construction; (ii) an Explorer-Augmented Leader Selection strategy that explicitly incorporates leaders from sparsely populated regions to improve guidance diversity; and (iii) an HV-triggered perturbation mechanism that activates only when progress stalls, injecting controlled diversity to escape local regions. In addition, FairWolf employs a persistent grid archive to stabilise density estimation across iterations and reduce unnecessary re-partitioning.

We next summarise the MOGWO template used as the starting point and then present the FairWolf mechanisms in detail. The control parameters used in MOGWO and the proposed FairWolf are summarised in Table 2.

5.1. Fundamentals of MOGWO

The Multi-Objective Grey Wolf Optimiser (MOGWO) is inspired by the hunting behaviour and social hierarchy of grey wolves [34]. Each wolf represents a candidate solution, and the hierarchy is divided into four levels: alpha (α), beta (β), delta (δ), and omega (ω). The first three wolves act as leaders, guiding the position update of the remaining wolves.

The mathematical model of the hunting strategy is expressed as follows. First, the distances of a wolf to the three leaders are computed:

$$\vec{D}_\alpha = \left|\vec{C}_1 \cdot \vec{X}_\alpha(t) - \vec{X}(t)\right|,$$

$$\vec{D}_\beta = \left|\vec{C}_2 \cdot \vec{X}_\beta(t) - \vec{X}(t)\right|,$$
$$\vec{D}_\delta = \left|\vec{C}_3 \cdot \vec{X}_\delta(t) - \vec{X}(t)\right|. \quad (12)$$

Then, three candidate positions are generated by following the leaders:

$$\vec{X}_1 = \vec{X}_\alpha(t) - \vec{A}_1 \cdot \vec{D}_\alpha,$$
$$\vec{X}_2 = \vec{X}_\beta(t) - \vec{A}_2 \cdot \vec{D}_\beta,$$
$$\vec{X}_3 = \vec{X}_\delta(t) - \vec{A}_3 \cdot \vec{D}_\delta. \quad (13)$$

Finally, the position of the wolf is updated as the average of the three candidate positions:

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}. \quad (14)$$

The coefficient vectors are defined as:

$$\vec{A} = 2 \cdot a \cdot \vec{r}_1 - a, \quad \vec{C} = 2 \cdot \vec{r}_2, \quad \vec{a} = 2 - \frac{2t}{MaxIt}. \quad (15)$$

where $\vec{X}(t)$ is the current position, $\vec{X}_\alpha(t)$, $\vec{X}_\beta(t)$, and $\vec{X}_\delta(t)$ are the three leaders, $\vec{D}_\alpha, \vec{D}_\beta, and\, \vec{D}_\delta$ are distance vectors between a wolf and each leader, and $\vec{r}_1, \vec{r}_2 \sim U(0,1)$. The coefficient $\vec{A}$ controls exploration/exploitation via its magnitude, while $\vec{C}$ stochastically scales the leader influence; a decreases linearly from 2 to 0 over iterations.

To extend GWO to multi-objective optimisation, MOGWO incorporates two additional mechanisms: the Grid-based Archive mechanism and the Leader Selection mechanism. The pseudo code of the MOGWO algorithm is provided in Algorithm 1.

Grid-based archive mechanism. An external archive is maintained to store non-dominated solutions [35]. For each new solution, dominance relations with archive members are checked: dominated solutions are discarded, dominating solutions replace weaker members, and mutually non-dominant solutions are added to the archive. The archive has a fixed maximum capacity, and when it reaches this limit, a grid-based truncation strategy is applied.

Following the Pareto Archived Evolution Strategy (PAES) [36], the objective space is divided into hypercubes (grid cells). Each archived solution is mapped to a cell according to its objective values, and the density of each cell is monitored. Once the archive reaches its maximum capacity, a truncation procedure is activated: the most crowded grid cell is identified, and one solution is randomly deleted from this cell to restore capacity. If a new solution lies outside the current grid boundaries, the grid is expanded to cover it, and all archive members are reassigned accordingly. This mechanism guarantees both a bounded archive size and the preservation of diversity along the Pareto front.

Leader selection mechanism. In single-objective GWO, α, β, and δ are simply the three best wolves. In MOGWO, leaders are instead selected from the archive using a roulette-wheel mechanism biased towards sparsely populated cells:

$$prob_i \propto \frac{c}{Num_i}, \quad (16)$$

where $c > 1$ is a constant and Num_i is the number of solutions in the ith grid cell. Consequently, solutions from less crowded regions of the Pareto front are more likely to be selected as leaders, thereby encouraging exploration of underrepresented areas.

In special cases, if the least crowded cell contains at least three solutions, they are randomly assigned to α, β, and δ; otherwise, leaders are supplemented from the next least crowded cells until three are obtained. This ensures that leaders do not cluster in the same region, helping to maintain a well-distributed approximation set.

Algorithm 1 Multi-objective grey wolf optimiser (MOGWO)

Input: pop_size, $MaxIt$, $archive\_size$, grid params $(\alpha_g, \beta_g, nGrid, \gamma_g)$
Output: $archive$ - Archive of non-dominated solutions
Initialise population $\{X_i\}_{i=1}^{pop\_size}$ and evaluate (f_1, f_2).
Initialise $archive$ with non-dominated solutions; build an adaptive grid in the objective space.
for $t = 1$ to $MaxIt$ **do**
    Select leaders $(X_\alpha, X_\beta, X_\delta)$ from sparse grid cells in $archive$ (grid-based roulette selection).
    Update coefficients $(a, \vec{A}, \vec{C})$ using Eq. (15).
    Update each wolf X_i using leader-guided rules (Eqs. (12) to (14)).
    Evaluate (f_1, f_2); update $archive$ by inserting new non-dominated solutions and removing dominated ones.
    if $archive$ exceeds capacity **then**
        Truncate $archive$ by deleting solutions from the most crowded grid cells until capacity is met.
    end if
    if new solutions exceed current grid bounds **then**
        Expand grid bounds and reassign all archive members to grid cells.
    end if
end for
return $archive$

5.2. The proposed FairWolf algorithm

In FairWolf, each wolf encodes a candidate solution in the discrete seed-set search space. Specifically, under the FMOIM budget constraint, the position of wolf i is represented as a feasible seed set $S^{(i)} \subseteq V$ with $|S^{(i)}| = k$. Algorithm 2 summarises the overall procedure. FairWolf consists of three components: (i) a discrete position updating operator for seed-set optimisation, (ii) an explorer-augmented leader selection strategy to improve guidance diversity, and (iii) an HV-triggered perturbation mechanism to mitigate stagnation.

Discrete position updating. Fig. 1 illustrates the proposed set-based discrete position updating in FairWolf through a running example with budget $k = 5$. At iteration t, wolf i encodes a feasible seed set $S^{(i)} \subseteq V$ with $|S^{(i)}| = k$, and a leader l is sampled from a leader pool $\mathcal{L}$ that combines (i) core leaders from the current non-dominated set and (ii) explorer leaders drawn from sparsely populated regions of the objective space (Section 5.2). The update modifies $S^{(i)}$ through node exchanges, thereby preserving $|S^{(i)}| = k$ by construction.

The position update for each wolf is implemented via set-based modifications rather than continuous movements. Given a current solution $S^{(i)}$ and a leader l, the update is determined by the parameter A: when $|A| < 1$, the wolf tends to align its node set with that of the leader (exploitation); when $|A| \geq 1$, it introduces differences from the leader's set (exploration). The parameter C controls the scale of modifications.

$$A = 2 \cdot a \cdot r_1 - a, \quad C = 2 \cdot r_2, \quad a = 2 - \frac{2t}{MaxIt}, \quad (17)$$

where A and C are scalar coefficients controlling the magnitude of set modifications, a decreases linearly from 2 to 0, and $r_1, r_2 \in [0,1]$ are random numbers sampled from a uniform distribution.

Exploitation Process: The wolf aligns its node set with that of the leader by exchanging nodes. The candidate nodes to be added and removed are defined in Eq. (18), and the number of replacements is determined by Eq. (19).

$$E_{\text{new}} = l \setminus S^{(i)}, \quad E_{\text{old}} = S^{(i)} \setminus l, \quad (18)$$

$$step\_exploit = \min\left(\lceil |A| \cdot n_{\text{swap}} \rceil, \lceil C \cdot n_{\text{swap}} \rceil\right),$$

$$n_{\text{swap}} = \min(|E_{\text{new}}|, |E_{\text{old}}|). \tag{19}$$

where n_{swap} denotes the maximum feasible number of node exchanges, determined by the smaller size of the candidate addition set E_{new} and the candidate removal set E_{old}.

Accordingly, *step_exploit* nodes are removed from E_{old} and replaced with nodes sampled from E_{new}.

Exploration Process: The wolf diverges from the leader by removing overlapping nodes by Eq. (20) and is substituted with nodes sampled from outside both sets by Eq. (21).

$$E_{\text{old}} = S^{(i)} \cap l, \tag{20}$$

$$E_{\text{cand}} \subseteq V \setminus (S^{(i)} \cup l). \tag{21}$$

where V is the ground set of all nodes. The update preserves the budget constraint $|S^{(i)}| = k$ by construction.

The number of exchanges is given by:

$$step\_explore = \min\big(\lceil |A| \cdot |E_{\text{old}}| \rceil,\ \lceil C \cdot |E_{\text{old}}| \rceil,\ |E_{\text{old}}|\big). \tag{22}$$

Thus, *step_explore* nodes are removed from E_{old} and substituted with nodes sampled from E_{cand}.

If the updated solution matches the original, random resampling is performed to maintain diversity within the population.

Explorer-augmented leader selection. In multi-objective optimisation, selecting leaders solely from the current non-dominated set may lead to *leader crowding*, in which guidance concentrates on already well-populated regions of the trade-off surface and reduces coverage. To address this, FairWolf adopts an explorer-augmented leader selection strategy that combines: (i) *core leaders* sampled from the current non-dominated archive to maintain convergence pressure, and (ii) *explorer leaders* sampled with a bias towards sparsely populated regions of the objective space to encourage exploration along underrepresented trade-off directions.

A persistent grid structure is employed to partition the objective space into *nGrid* cells per dimension. Unlike the adaptive grids in classical MOGWO, the persistent grid expands only when new solutions exceed the current bounds, thereby avoiding frequent re-partitioning and stabilising density estimation across iterations. Since both objectives are normalised to $[0, 1]$, the grid stabilises after a limited number of expansions, enabling consistent density estimation while reducing redundant computation. When the archive exceeds its maximum capacity, solutions in overcrowded cells are pruned based on density and crowding-distance criteria.

The probability of selecting a grid cell i for explorer leaders is defined as:

$$prob_i \propto (Num_i + \epsilon)^{-\beta_g}, \tag{23}$$

where Num_i denotes the number of solutions in grid cell i, ϵ is a small constant to avoid division by zero, and β_g controls the strength of the bias towards sparse cells. Higher values of β_g increase the selection bias towards sparse regions.

Once a cell is selected, candidate solutions within that cell are ranked by their crowding distance [32], and the solution with the largest crowding distance is chosen as the explorer leader.

HV-triggered perturbation mechanism. To mitigate stagnation, FairWolf employs a hypervolume (HV)-triggered perturbation strategy. HV is used as an indicator of the coverage and quality of the current archive [37]. Rather than applying unconditional random restarts that discard accumulated search information, the perturbation is activated only when the archive shows insufficient progress within a short time window, thereby avoiding unnecessary disruption during effective search phases. At iteration t, we compute $HV(t)$ from the current archive and maintain a sliding window of length τ. Let

$$\mathcal{W}(t) = \{HV(t - \tau + 1), \ldots, HV(t)\}. \tag{24}$$

Algorithm 2 FairWolf Algorithm

Input: Graph $G = (V, E)$, Population *pop* and its size *pop_size*, Budget k, *archive size*, $MaxIt$, Number of explorer leaders $n_{\text{explorers}}$, HV window size τ, HV stagnation threshold ϵ_{HV}, Perturbation ratio *perturb*, grid params $(\alpha_g, \beta_g, nGrid, \gamma_g)$

Output: *archive* - set of non-dominated solutions

Initialise population $pop = \{S^{(i)}\}_{i=1}^{pop\_size}$ with random seed sets of size k.
Evaluate each $S^{(i)}$ with objectives (f_1, f_2).
Initialise *archive* with non-dominated solutions.
Construct persistent grid structure with parameters $(\alpha_g, \beta_g, nGrid, \gamma_g)$.
$t \leftarrow 0$
while $t < MaxIt$ **do**
  Update control parameter a, A, and C by Eq. (17).
  Select leader set $\mathcal{L}$ from *archive* (core + $n_{\text{explorers}}$ explorer leaders). *(Explorer-Augmented Leader Selection)*
  for each wolf $S^{(i)} \in pop$ **do**
    Choose a leader $l \in \mathcal{L}$ uniformly at random.
    Update $S^{(i)}$ by discrete set-based rules using Eqs. (18) to (22). *(Position Updating)*
    If $S^{(i)}$ unchanged, re-sample randomly to maintain diversity.
    Evaluate $S^{(i)}$ with (f_1, f_2).
  end for
  Update *archive* with new non-dominated solutions.
  Compute hypervolume $HV(t)$.
  if hypervolume stagnation over window τ $(\Delta HV < \epsilon_{HV})$ **then**
    Apply perturbation to a randomly selected fraction of the population (ratio = *perturb*). *(HV-triggered Perturbation)*
  end if
  $t \leftarrow t + 1$
end while
return *archive*

We measure the progress within the window by the HV range:

$$\Delta HV(t) = \max \mathcal{W}(t) - \min \mathcal{W}(t). \tag{25}$$

Perturbation is triggered if

$$\Delta HV(t) < \epsilon_{HV}, \tag{26}$$

where ϵ_{HV} is a small threshold.

Once triggered, perturbation is applied to a randomly selected fraction of the population, controlled by *perturb*. For each selected solution $S^{(i)}$, we perform a single swap to preserve the budget constraint:

$$S'^{(i)} = (S^{(i)} \setminus \{u\}) \cup \{v\}, \tag{27}$$

where u is sampled uniformly from $S^{(i)}$ and v is sampled from $V \setminus S^{(i)}$. The perturbed solutions are then re-evaluated and inserted back into the population (and subsequently into the archive update routine). By activating only under HV stagnation, this operator injects diversity when the search is trapped, while keeping the search stable when HV is improving.

6. Experimental results and discussion

6.1. Experiment settings

6.1.1. Datasets

In this study, we employ eight real-world datasets to evaluate the effectiveness of the proposed method. All networks are undirected, and their community structures are identified using the Leiden algorithm [38]. This algorithm is chosen for its efficiency and robustness in detecting well-connected communities based on the underlying topological structure of the networks. A summary of the network characteristics is presented in Table 3.

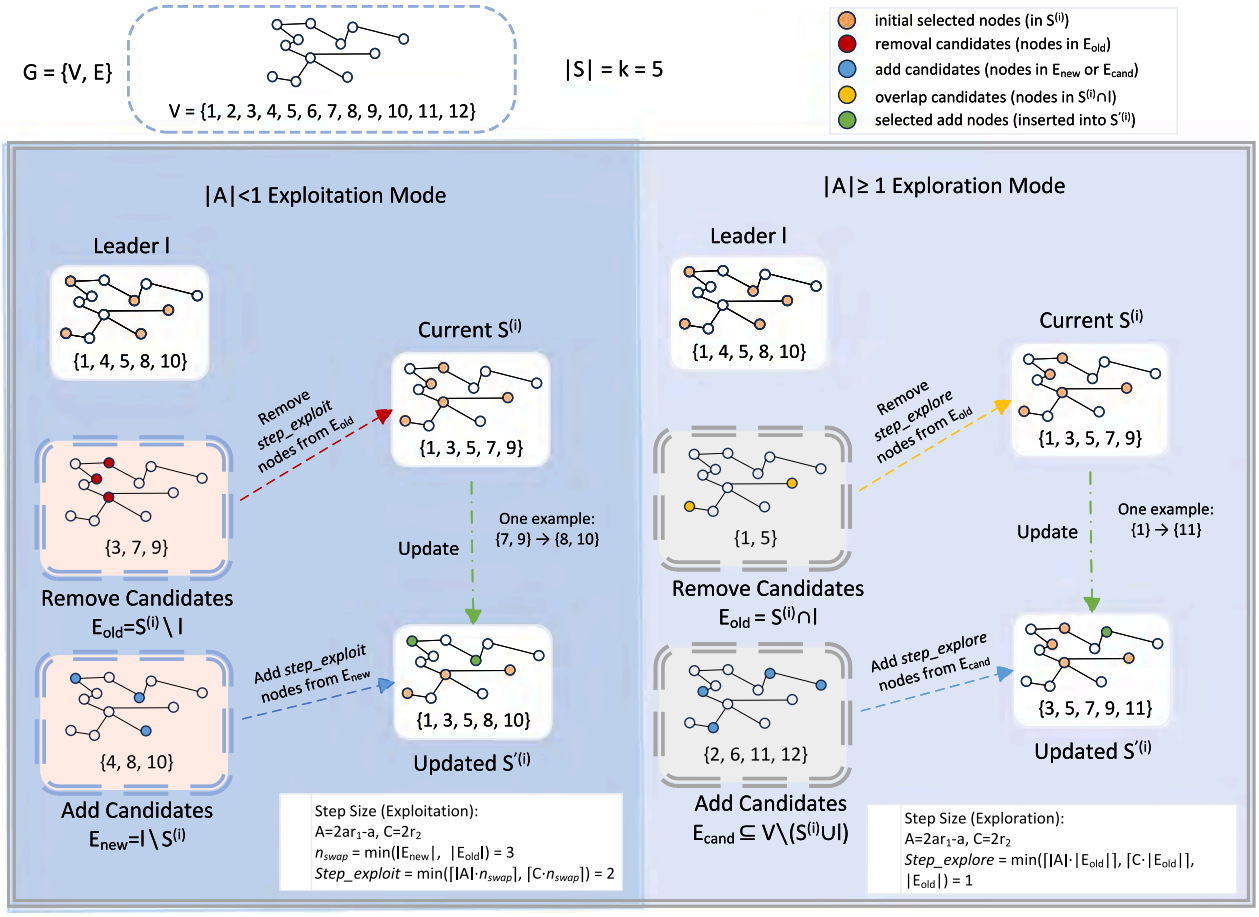


Fig. 1. Illustration of the discrete position updating process in FairWolf.

Table 3
Topological properties of experimental networks.

Networks	$\|V\|$	$\|E\|$	d_{max}	d_{min}	d_{avg}	ACC	NOC
Dolphin Network (Dolphin)	62	159	12	1	5	0.2590	4
FB-Pages-Food network (Food)	620	2102	134	1	6.7806	0.3309	11
Email-Eu-Core network (Email)	1005	25,571	347	1	33.2458	0.3994	8
Hamsterster network (Ham)	2426	16,630	273	1	13.7098	0.5375	25
Facebook network (Facebook)	4039	88,234	1045	1	43.6910	0.6055	17
Rice31 network (Rice)	4087	184,828	581	1	90.4468	0.2942	10
Wikipedia Vote network (Wiki)	7115	103,689	1065	1	28.3238	0.1409	4
LastFM Asia Social Network (LastFM)	7624	27,806	216	1	7.2943	0.2194	18

Specifically, $|V|$ and $|E|$ denote the total number of nodes and edges, respectively; d_{max} represents the maximum degree; d_{min} represents the minimum degree; d_{avg} is the average degree; ACC indicates the average clustering coefficient; NOC denotes the number of communities detected by Leiden algorithm.

- Dolphin [39]: A small social network of 62 bottlenose dolphins, where 159 edges represent frequent associations between dolphins. The network is moderately cohesive and divided into 4 natural communities.
- FB-Pages-Food Network [40]: This network was collected from Facebook pages in November 2017, where nodes represent pages and edges denote mutual "likes". The network contains 620 nodes and 2102 edges. It is characterised by a small, sparse structure and is partitioned into 11 distinct communities.
- Email-Eu-Core Network [41,42]: Derived from email communications within a large European research institution, this network models individuals as nodes and email exchanges as edges. It features a moderate clustering coefficient of 0.3994 and is partitioned into 8 communities of roughly comparable sizes.
- Hamsterster Network [40]: This social network captures friendship and familial relationships among users of the Hamsterster platform. It comprises 2426 nodes, 16,630 edges, and is organised into 25 relatively small communities.
- Facebook Network [43]: This network was obtained through surveys administered to Facebook app users. It is structured into 17 distinct communities, reflecting social groupings based on the responses.
- Rice31 Network [40]: Extracted from Facebook friendship data, this network represents individuals as nodes and social ties as edges. It is a highly dense network with an average degree of 90.4 and is divided into 10 communities that reflect different student groups.
- Wikipedia Vote Network [44,45]: This network contains data on voting behaviour in Wikipedia's admin elections, spanning from the platform's inception to January 2008. It includes 7115

users and 103,689 voting links, resulting in 4 highly imbalanced communities.

- LastFM Asia Social Network [46]: A social network of 7624 LastFM users from Asian countries, collected from the public API in March 2020. Nodes represent users, edges indicate mutual follower relationships, and node features are derived from the artists that users like. The network is divided into 18 communities.

6.1.2. Baseline algorithms and parameter setting

To evaluate FairWolf, we benchmark it against two groups of baselines: (i) multi-objective algorithms tailored for FMOIM, including GFMOGWO [27], MODBA [47], MODPSO [48,49], and SetMOGWO [24]; and (ii) canonical multi-objective evolutionary algorithms, namely MOEA/D [50] and SPEA2 [51].

- GFMOGWO [27]: The Group-Fair-awareness Multi-objective Grey Wolf optimiser (GFMOGWO) is one of our previous works, published in 2024. This study extends the classical Multi-objective Grey Wolf optimiser (MOGWO) proposed by Mirjalili [34] to address the dual discrete objectives of maximising influence and ensuring fairness.
- MODBA [47]: The Multi-Objective Discrete Bat Algorithm (MODBA) is designed for discrete multi-objective optimisation problems. Inspired by the echolocation behaviour of bats, it incorporates turbulence operations and mutation strategies to enhance population diversity and search effectiveness.
- MODPSO [48,49]: The Multi-objective Discrete Particle Swarm Optimisation (MODPSO) algorithm extends the classical MOPSO framework [48] by integrating elements of the Discrete PSO algorithm [49]. It is specifically adapted to address discrete multi-objective problems such as fair influence maximisation.
- SetMOGWO [24]: The SetMOGWO is a multi-objective Fair IM algorithm based on the Grey Wolf Optimiser, employing a set-based representation for solution encoding and evolution. In each iteration, it randomly selects a leader and updates individuals by adding or deleting nodes to move towards or away from the leader. An adaptive grid-based archive is used to maintain the diversity of the solution set.
- MOEA/D [50]: MOEA/D is a canonical decomposition-based multi-objective evolutionary algorithm. It decomposes a multi-objective problem into a set of scalar subproblems using weight vectors and optimises them collaboratively by maintaining one solution per subproblem. Neighbouring subproblems share information through local variation and replacement, which promotes both convergence and diversity.
- SPEA2 [51]: SPEA2 is a strength Pareto evolutionary algorithm that employs an external archive to store non-dominated solutions. It assigns each solution a fitness value based on Pareto strength and density estimation, and then performs selection from the union of the current population and the archive. When the archive exceeds its capacity, a truncation procedure based on pairwise distances is used to preserve diversity.

Each experimental run is executed for 100 iterations, and each experiment is independently repeated 10 times to ensure statistical reliability. The propagation probabilities $p = 0.01, 0.05, 0.1$ are selected to represent low, moderate, and relatively stronger diffusion regimes, respectively. These values are used to examine whether the proposed framework remains effective under different propagation intensities across heterogeneous networks. For a fair comparison, the common parameters shared by FairWolf, SetMOGWO, and GFMOGWO (i.e., α_g, β_g, $nGrid$, γ_g, a, r_1, and r_2) are assigned the same values as those specified in the original MOGWO paper [34]. A detailed summary of the experimental parameters is provided in Table 4.

6.1.3. Computational complexity

Let T be the number of iterations (MaxIt), n the population size (pop_size), m the archive size upper bound (archive_size), and k the seed budget. Let f_{eval} denote the cost of evaluating one seed set (i.e., Monte Carlo diffusion with N simulations and the corresponding fairness computations).

FairWolf. Each iteration of FairWolf contains four main steps. (i) *Discrete position updating*: each wolf is updated by set-based swaps and sampling while preserving $|S| = k$. The number of modified elements is bounded by k, yielding $O(nk)$ for updating n wolves in typical cases. (ii) *Objective evaluation*: each updated seed set is evaluated once, contributing $O(n\, f_{\text{eval}})$. (iii) *Archive maintenance*: the current population and archive are merged (size at most $n+m$), and only the first non-dominated front is retained; when the archive exceeds capacity, grid-based truncation is applied. The dominance-based filtering has a worst-case cost of $O((n+m)^2)$. (iv) *HV monitoring*: the bi-objective hypervolume of the archive is computed for stagnation detection. For two objectives, HV can be computed efficiently after sorting, which costs $O(m \log m)$. The sliding-window stagnation check additionally scans a window of length τ and costs $O(\tau)$ per iteration, where τ is a small constant.

Combining these terms gives

$$O(\text{FairWolf}) = O\Big(T \cdot \Big[n \cdot (k + f_{\text{eval}}) + (n+m)^2 + m \log m\Big]\Big). \tag{28}$$

Baselines. SetMOGWO follows the same archive-based discrete search structure as FairWolf (set-based updates, evaluation, and archive maintenance), and thus has the same leading-order terms without the HV monitoring cost:

$$O(\text{SetMOGWO}) = O\Big(T \cdot \Big[n \cdot (k + f_{\text{eval}}) + (n+m)^2\Big]\Big). \tag{29}$$

GFMOGWO introduces additional graph-wide processing (e.g., ranking or scoring over all nodes), which adds a dependence on $|V|$:

$$O(\text{GFMOGWO}) = O\Big(T \cdot \Big[n \cdot (|V| \log |V| + f_{\text{eval}}) + (n+m)^2\Big]\Big). \tag{30}$$

For MODPSO, particle-based update and repair commonly involve node-level processing over V (e.g., constructing scores/probabilities and selecting k nodes), giving

$$O(\text{MODPSO}) = O\Big(T \cdot \Big[n \cdot (|V| + f_{\text{eval}}) + (n+m)^2\Big]\Big). \tag{31}$$

MODBA further employs neighbourhood-based local search; if each local move examines d_{avg} neighbours per selected seed, its per-solution cost includes an additional $O(d_{\text{avg}} k)$ term:

$$O(\text{MODBA}) = O\Big(T \cdot \Big[n \cdot (|V| + d_{\text{avg}} k + f_{\text{eval}}) + (n+m)^2\Big]\Big). \tag{32}$$

MOEA/D maintains one solution per weight vector (hence n = pop_size) and performs neighbourhood-based replacement with neighbourhood size T_{nb}; each offspring generation costs $O(k + f_{\text{eval}})$ and is compared against at most T_{nb} neighbours, with additional archive dominance checks bounded by $O(m)$:

$$O(\text{MOEA/D}) = O\big(T \cdot n \cdot \big[(k + f_{\text{eval}}) + T_{\text{nb}} + m\big]\big). \tag{33}$$

SPEA2 computes strength-based fitness and density estimation over the union set U of size $|U| = n + m$. Pairwise dominance checks cost $O(|U|^2)$, and kth nearest-neighbour density estimation requires pairwise distances plus sorting, resulting in $O(|U|^2 \log |U|)$ in the worst case:

$$O(\text{SPEA2}) = O\Big(T \cdot \Big[|U|^2 \log |U| + n \cdot (k + f_{\text{eval}})\Big]\Big). \tag{34}$$

For FMOIM, the evaluation cost f_{eval} (Monte Carlo diffusion and fairness computation) is typically the dominant component across methods. Among algorithmic overheads, archive-based approaches share the same principal term $O((n+m)^2)$ due to dominance filtering and truncation. In this context, HV monitoring in FairWolf is constrained by a fixed archive size and is efficient in the bi-objective case. To complement the asymptotic analysis, we additionally report empirical running time (mean±std over 10 runs) in Tables 8 to 10 and discuss it in Section 6.3.2.

Table 4
Experimental settings and algorithm-specific parameters.

Common experimental settings	Values
Max iterations ($MaxIt$)	100
Independent runs	10
Seed budget (k)	30
Activation probability (p)	0.01, 0.05, 0.1
Monte Carlo simulations per evaluation (N)	10
Archive capacity ($archive\_size$)	100
Population size (pop_size)	100
HV reference point	(0, 0)
Grid-archive settings (used by FairWolf, SetMOGWO, GFMOGWO)	**Values**
Grid inflation (α_g)	0.1
Leader selection pressure (β_g)	4
Grids per dimension ($nGrid$)	10
Density-deletion parameter (γ_g)	2
Control parameter (a)	linearly decreased from 2 to 0
Random vectors (r_1, r_2)	Randomly Selected in $[0, 1]$
FairWolf-specific settings	**Values**
Number of explorer leaders ($n_{\text{explorers}}$)	2
HV window size (τ)	5
Perturbation ratio ($perturb$)	0.15
Equity weight (λ)	0.5
MODBA settings	**Values**
Initial pulse rate/maximum pulse rate	0.1/0.9
Minimum frequency/maximum frequency	0.5/1.5
Initial loudness/minimum loudness	1/0.1
Loudness reduction factor	0.9
Pulse rate growth factor	0.9
MOEA/D settings	**Values**
Decomposition	Tchebycheff
Neighbourhood size T_{nb}	10
Weight vectors	evenly spaced in 2D, $\lvert W\rvert = pop\_size$
Mutation probability	0.2
SPEA2 settings	**Values**
External archive size	$archive\_size$
Density estimator	kth nearest neighbour
Mutation probability	0.2

6.1.4. Evaluation metrics

To evaluate the proposed FairWolf algorithm and the baseline methods, we use several widely applied performance indicators. These include the Pareto Front (PF) [31], Hypervolume (HV) [37], Inverted Generational Distance (IGD) [25,52], Spacing (SP) [53], Spread (Δ) [54], Computational Time, and two fairness-related measures, the Price of Fairness (POF) and the Price of Influence (POI) [11,25].

- Pareto Front (PF): The Pareto front refers to the set of non-dominated optimal solutions. It provides a reference for evaluating the closeness of the approximated solutions to the true trade-off surface.
- Hypervolume (HV): The hypervolume measures the size of the objective space dominated by the obtained solutions with respect to a reference point. In this study, the reference point is set at $(0, 0)$. Larger HV values indicate better convergence and diversity.

$$HV = \int_{\mathbf{r}}^{\mathbf{PF}} dV \tag{35}$$

where **r** is the reference point and **PF** is the non-dominated Pareto front.

- Inverted Generational Distance (IGD): IGD evaluates the average distance from a set of uniformly distributed points on the true Pareto front (PF_{true}) to the nearest solutions in the approximated front (PF_{approx}). Smaller IGD values imply better convergence and coverage. This indicator complements fairness-specific measures by evaluating the convergence and distribution quality of the obtained Pareto front, ensuring that fairness improvements do not come at the cost of degraded optimisation performance.

$$IGD = \frac{1}{|PF_{\text{true}}|} \sum_{p \in PF_{\text{true}}} \min_{x \in PF_{\text{approx}}} d(p, x) \tag{36}$$

where PF_{true} represents the true Pareto front. PF_{approx} denotes the approximated Pareto front obtained by the algorithm. $d(p, x)$ is the Euclidean distance between a true Pareto front point p and its closest point x in the approximated Pareto front.

However, as stated above, computing IGD requires knowledge of the true Pareto front (PF_{true}), which is often difficult to obtain for most real-world multi-objective problems (MOPs), including the FMOIM problem addressed in this paper. To overcome this challenge, we adopt an approximation strategy inspired by previous studies [25, 30,55]. Specifically, we first merge the Pareto solutions obtained from all algorithms into a unified Pareto set (PS'). Next, we apply the Pareto dominance principle to extract the final set of non-dominated solutions (PS) from PS'. This refined PS serves as the approximate true Pareto front, enabling us to compute the IGD value for each algorithm effectively.

- Spacing (SP): The spacing indicator measures the uniformity of distances between neighbouring solutions on the Pareto front. A smaller SP value indicates that the solutions are more evenly distributed.

$$SP(Y_N) = \sqrt{\frac{1}{|Y_N| - 1} \sum_{j=1}^{|Y_N|} \left(\overline{d} - d^1(y^j, Y_N \setminus \{y^j\})\right)^2} \tag{37}$$

where

$$d^1(y^j, Y_N \setminus \{y^j\}) = \min_{y \in Y_N \setminus \{y^j\}} \|y - y^j\|_1 \tag{38}$$

is the L_1 distance from solution $y^j \in Y_N$ to its nearest neighbour, and $\overline{d}$ is the average of all nearest neighbour distances.

- Spread (Δ): Spread evaluates how widely the non-dominated solutions are distributed along the Pareto front. Smaller values of Δ show better coverage and more uniform distribution.

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{NPF} \left| d_i - \overline{d} \right|}{d_f + d_l + (NPF - 1)\overline{d}} \tag{39}$$

where d_f and d_l denote the Euclidean distances between the extreme solutions (first and last points) of PF_{optimal} and PF_g, respectively. d_i is the Euclidean distance between the ith solution in PF_g and the nearest solution in PF_{optimal}, $\overline{d}$ is the mean of all d_i, and NPF is the number of non-dominated solutions in PF_g.

When $\Delta = 0$, all extreme points of PF_{optimal} are included, representing an ideal coverage of the Pareto front.

- Price of Fairness (POF) and Price of Influence (POI):
  POF and POI are used to examine the trade-off between two objectives. Let $\sigma_{\max}$ and $\sigma_{\min}$ denote the maximum and minimum influence spread values of objective f_1. Let $F_{\max,i}$ and $F_{\min,i}$ denote the maximum and minimum values of the fairness objective f_2.

$$\text{POF} = \frac{\sigma_{\max} - \sigma_{\min}}{\sigma_{\max}}, \tag{40}$$

$$\text{POI} = \frac{F_{\max} - F_{\min}}{F_{\max}}. \tag{41}$$

A smaller POF means that less influence is lost when fairness is improved. A smaller POI means that fairness is satisfied at a lower cost to influence.

6.2. Experiment 1: Ablation study and parameter sensitivity analysis

For both the ablation study and the parameter sensitivity analysis, experiments were conducted on the FB-Pages-Food Network [40] with a population size of $pop = 100$ and a fixed number of iterations, $T = 100$. The budget was fixed at $k = 30$, and the activation probability was set to $p = 0.1$. Each configuration was executed 10 times independently, and the results were averaged. The evaluation considered five performance indicators: Hypervolume (HV) [37], Inverted Generational Distance (IGD) [25,52], Spacing (SP) [53], Spread (Δ) [54], and Computational Time (Time).

6.2.1. Ablation study

To assess the contributions of the two add-on mechanisms in our proposed FairWolf algorithm, we conduct an ablation study by progressively removing them: the HV-triggered Perturbation Mechanism and the Explorer-Augmented Leader Selection Mechanism. The number of explorer leaders $n_{\text{explorers}}$ and the perturbation ratio $perturb$ are fixed at 2 and 0.15, respectively. Three algorithmic variants are evaluated:

- **FMODGWO**: baseline version without any additional mechanisms;
- **H-FMODGWO**: incorporates only the HV-triggered Perturbation Mechanism;
- **FairWolf(EH-FMODGWO)**: full version with both mechanisms integrated.

Table 5 reports the quantitative results, and Fig. 2 shows the HV convergence curves (mean ± Std over 10 runs).

Overall, H-FMODGWO improves the optimisation trajectory over the baseline FMODGWO, while incorporating both mechanisms (FairWolf) yields the best front quality and stability. In particular, FairWolf achieves the highest HV and the lowest IGD, together with improved SP, Δ and reduced runtime (Table 5). From the convergence curves in Fig. 2, HV-triggered perturbation mainly accelerates early progress, whereas the explorer-augmented leader selection further improves the search coverage and stabilises convergence, leading to a higher final HV.

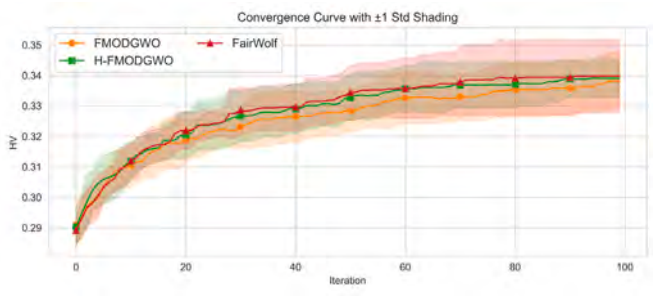


Fig. 2. HV-based convergence curves (mean ± Std) of the ablation variants on the Food network.

6.2.2. Parameter sensitivity analysis

We examine the sensitivity of FairWolf to key design choices from two perspectives: (i) search dynamics, and (ii) the construction of the equity fairness objective. All results are summarised in Tables 6 and 7.

Sensitivity to search parameters. We first investigate two key search parameters: the number of explorers and the perturbation ratio. Each parameter was examined at three levels, namely $n_{\text{explorers}} \in \{0, 1, 2\}$ and $perturb \in \{0.05, 0.10, 0.15\}$. This resulted in a full factorial design with $3 \times 3 = 9$ configurations ($S1$–$S9$). The detailed performance of each configuration is reported in Table 6.

Overall, $S9$ ($n_{\text{explorers}} = 2$, $perturb = 0.15$) provides a favourable balance between Pareto-front quality and computational cost, achieving relatively higher HV and lower IGD. Therefore, it is adopted as the default setting in the subsequent experiments. Once selected, this configuration is kept unchanged across all networks and propagation probabilities in the main experiments. This setting provides a consistent basis for evaluating the generalisation ability of FairWolf across heterogeneous network structures and diffusion conditions, while reducing the risk of dataset-specific overfitting.

Sensitivity to the fairness weight λ in objective 2. With the default search configuration fixed, we further assess the robustness of the equity fairness objective with respect to the weighting parameter λ, which balances two complementary fairness components: the Jensen–Shannon Divergence similarity $JS_{\text{sim}}(P, Q)$ and Jain's fairness index $Jain(P/Q)$, as defined in Eq. (8). These metrics provide complementary fairness interpretations: prioritising global alignment alone may still permit local proportional disparities, while focusing solely on proportional consistency may tolerate systematic shifts of the overall distribution away from the population structure. The weighted combination in (8), therefore, explicitly controls this trade-off through λ.

In this study, all other algorithmic settings are fixed to the default configuration ($n_{\text{explorers}} = 2$, $perturb = 0.15$), and only λ is varied with three representative settings: Fair1 ($\lambda = 0.3$), Fair2 ($\lambda = 0.5$), and Fair3 ($\lambda = 0.7$). As reported in Table 7, Fair2 achieves the best overall trade-off, obtaining the highest hypervolume ($HV = 0.3422 \pm 0.0054$) while maintaining a low IGD (0.0083 ± 0.0032), together with improved spacing (Spacing = 0.001194245). Fair3 yields a slightly lower IGD, a slightly lower HV, and a higher runtime. Overall, $\lambda = 0.5$ provides a stable and favourable balance between the two fairness components, and is adopted as the default setting in subsequent experiments.

Table 5
Ablation study results.

Algorithms	IGD ($\mu \pm \delta$)	HV ($\mu \pm \delta$)	Spacing	Spread	Time ($\mu \pm \delta$)
FMODGWO	0.0123 ± 0.0054	0.3382 ± 0.0098	0.0020	0.02398	0.2572 ± 0.0159
H-FMODGWO	0.0114 ± 0.0035	0.3391 ± 0.0067	0.0029	0.01314	0.3058 ± 0.0118
FairWolf(EH-FMODGWO)	**0.0111 ± 0.0052**	**0.3398 ± 0.0126**	**0.0018**	**0.0050**	**0.2238 ± 0.0679**

Table 6
Sensitivity analysis of $n_{\text{explorers}}$ and *perturb* (S1–S9).

ID	$n_{\text{explorers}}$	*perturb*	IGD ($\mu \pm \delta$)	HV ($\mu \pm \delta$)	Spacing	Spread	Time ($\mu \pm \delta$)
S1	0	0.05	0.0119 ± 0.0053	0.3391 ± 0.0070	0.0011	**0.0203**	0.3093 ± 0.0114
S2	1	0.05	0.0115 ± 0.0057	0.3403 ± 0.0093	0.0012	0.0297	0.2705 ± 0.0588
S3	2	0.05	0.0107 ± 0.0052	0.3404 ± 0.0072	**0.0010**	0.0329	**0.1648 ± 0.0043**
S4	0	0.1	0.0123 ± 0.0051	0.3385 ± 0.0085	0.0026	0.0300	0.3171 ± 0.0177
S5	1	0.1	0.0116 ± 0.0039	0.3379 ± 0.0059	0.0013	0.0348	0.3220 ± 0.0462
S6	2	0.1	0.0109 ± 0.0042	0.3394 ± 0.0077	0.0013	0.0288	0.3271 ± 0.0090
S7	0	0.15	0.0113 ± 0.0047	0.3412 ± 0.0070	0.0015	0.0450	0.1759 ± 0.0047
S8	1	0.15	0.0121 ± 0.0039	0.3385 ± 0.0054	0.0015	0.0357	0.1919 ± 0.0145
S9	2	0.15	**0.0085 ± 0.0036**	**0.3447 ± 0.0078**	0.0011	0.0310	0.2025 ± 0.0047

Table 7
Fairness-weight sensitivity of λ (Fair1–Fair3).

ID	λ	IGD ($\mu \pm \delta$)	HV ($\mu \pm \delta$)	Spacing	Spread	Time ($\mu \pm \delta$)
Fair1	0.3	0.0093 ± 0.0044	0.3387 ± 0.0081	0.001654935	0.004331294	**0.1725 ± 0.0320**
Fair2	0.5	0.0083 ± 0.0032	**0.3422 ± 0.0054**	**0.001194245**	0.001591407	0.2066 ± 0.0507
Fair3	0.7	**0.0072 ± 0.0030**	0.3411 ± 0.0060	0.001408903	**0.001347406**	0.2478 ± 0.0534

6.3. Experiment 2: Comparison with multi-objective IM algorithms

To evaluate the effectiveness of the proposed FairWolf, we compare it with several representative multi-objective algorithms across eight benchmark networks. The evaluation is carried out under three different propagation probabilities ($p \in \{0.01, 0.05, 0.1\}$) to assess performance under varying diffusion intensities. Five metrics are considered: Hypervolume (HV) [37], Inverted Generational Distance (IGD) [25,52], Spacing (SP) [53], Spread (Δ) [54], and Computational Time (Time). All results are averaged over 10 independent runs.

6.3.1. Pareto front quality

Tables 8 to 10 report the quantitative results under $p \in \{0.01, 0.05, 0.1\}$, and Figs. 3 to 5 provide complementary evidence from convergence behaviour and objective-space distributions.

FairWolf shows the strongest overall Pareto-front quality under the low propagation probability setting ($p = 0.01$). It achieves the best HV on five datasets, indicating consistently better coverage of the trade-off surface. Meanwhile, FairWolf attains the best IGD on *Wiki*, *LastFM*, and *Facebook*, and remains very close to the best-performing IGD method on *Rice* and *Ham*, with only marginal differences. These results suggest that the improvements in front coverage are largely aligned with proximity to the reference trade-off surface, rather than being obtained by sacrificing convergence. This advantage is particularly evident at $p = 0.01$, where diffusion signals are weaker, and the search is more prone to stagnation. This observation is consistent with the design goal of FairWolf, which strengthens exploration while retaining convergence guidance. Its SP is generally mid-range, while Δ reaches the best value on three datasets and stays close to the best on the remaining ones.

Fig. 3 further reveals differences in search dynamics under $p = 0.01$. On most datasets, FairWolf improves HV rapidly in the first 20 to 40 iterations and continues to make steady progress afterwards, whereas several baselines exhibit an earlier plateau. This indicates that FairWolf's advantage is not only in the final HV values, but also in maintaining effective search throughout the optimisation process. Moreover, the shaded regions in Fig. 3 report the standard deviations over independent runs, providing an indication of the variability of the search process. In the later iterations, the HV trajectories remain within a relatively stable range for most algorithms. SPEA2 shows more visible generation-to-generation oscillations in several cases, while FairWolf exhibits comparatively smoother late-stage trajectories.

This behaviour is consistent with archive-based multi-objective optimisation, where newly generated non-dominated solutions may update the external archive and locally adjust the distribution of the approximated Pareto front. For FairWolf, the persistent grid archive and HV-triggered perturbation mechanism help maintain diversity while reducing premature stagnation. As a result, FairWolf maintains active search progress in the early and middle stages and remains relatively stable near the terminal generations. Together with the mean and standard deviation values reported in Tables 8 to 10, these results indicate stable search behaviour across repeated runs.

As the propagation probability increases to $p = 0.05$, FairWolf remains consistently competitive. It obtains the best HV on four datasets, and ranks near the top on the remaining networks (typically second or third), with small gaps to the best results. A similar pattern is observed for IGD, where FairWolf achieves the best values on two datasets and remains in the mid-to-top range elsewhere. For SP, FairWolf attains the best performance on two datasets and stays close to the best on the remaining ones, while Δ is generally ranked second, reflecting stable front extent across networks. In this setting, MOEA/D and SPEA2 frequently appear among the strongest competitors. However, when either baseline achieves the best performance on a dataset, FairWolf remains within the top-ranked group, indicating no pronounced performance degradation across networks.

Under the highest propagation probability setting ($p = 0.1$), FairWolf continues to achieve the best HV on four datasets. MOEA/D becomes the closest competitor in this setting and attains the best HV on three datasets, while FairWolf's IGD results are mostly ranked second, suggesting that its obtained fronts remain close to the reference set even under more intensive propagation. The SP and Δ results are generally mid-range, which is consistent with the reduced discrimination of distributional indicators when propagation is strong. In this setting, stronger propagation reduces evaluation noise and makes convergence-oriented strategies more competitive, which explains the narrower gaps among leading baselines.

We summarise the results using an average ranking analysis in Table 11. For each propagation probability, algorithms are ranked on each

dataset according to each Pareto-front quality indicator. A larger value is preferred for HV, whereas smaller values are preferred for IGD, SP, and Δ. The average rank is then computed across all datasets and indicators under the same propagation probability, and the overall rank is obtained by averaging across the three probability settings. A smaller rank indicates better overall performance.

As shown in Table 11, FairWolf obtains the best overall average rank among the compared algorithms. It ranks first under $p = 0.01$ and $p = 0.05$, and remains competitive under $p = 0.1$, where MOEA/D achieves the best average rank. This result indicates that FairWolf does not rely on isolated metric-level wins, but provides strong overall Pareto-front quality across heterogeneous networks and propagation settings.

Fig. 4 compares FairWolf solutions with a reference (True) Pareto front under $p = 0.01$. The reference front is constructed by pooling the final Pareto-optimal sets from all compared algorithms and extracting the non-dominated subset from their union. In this comparison, strong performance is characterised by solutions that remain close to the reference trade-off curve while covering a wide range along the spread-fairness trade-off. Fig. 5 further shows that FairWolf typically produces a broader and more evenly populated set of non-dominated solutions, whereas some baselines concentrate on limited regions of the objective space.

To provide an interpretable view of fairness beyond Pareto-front indicators, we examine community-level allocation bias on the Ham network under $p = 0.1$. For a seed set, the evaluator returns the realised activation-share distribution $P = \{P_j\}$ and the population-share reference $Q = \{Q_j\}$ (defined in Section 4.3). We define the signed allocation bias as $\delta_j = P_j - Q_j$, where $\delta_j > 0$ indicates over-allocation and $\delta_j < 0$ indicates under-allocation.

We summarise the worst community mismatch of a solution by the *Worst-case Allocation Deviation*

$$WAD(s) = \max_j |\delta_j| = \max_j |P_j - Q_j|. \tag{42}$$

For each run, we compute $WAD(s)$ for all solutions in the final Pareto set and report the PF-set median as a robust run-level summary. Fig. 6 shows that MOEA/D achieves the lowest median WAD, while FairWolf is a close second with a compact interquartile range, indicating consistently small worst-case deviations across runs. Fig. 7 visualises the PF-set median signed deviation vector δ_j, obtained by taking the median of δ_j over Pareto solutions within a run. FairWolf exhibits a more balanced pattern with most communities staying near zero deviation and no pronounced over-/under-allocation bands, whereas several baselines show stronger positive/negative patches concentrated on a subset of communities.

6.3.2. Runtime analysis

Tables 8 to 10 report the average runtime per run (mean±Std) under different propagation probabilities. Overall, the runtime of all algorithms increases with p, because the diffusion evaluation cost f_{eval} becomes higher when more activations occur. This effect is shared by all methods, since diffusion simulation dominates the evaluation stage.

Across datasets, FairWolf exhibits moderate runtime and is comparable to other set-based archive-driven optimisers (e.g., SetMOGWO). This observation is consistent with the complexity analysis. FairWolf has the same dominant cost components as the baselines, namely the population-level evaluation term $n \cdot f_{\text{eval}}$ and archive maintenance, whereas its additional HV-stagnation check only scans a short window and is activated only occasionally. By contrast, SPEA2 can be slower on several datasets, consistent with its fitness assignment and density estimation performed on the union of the population and archive $(n + m)$. MOEA/D typically shows moderate runtime, as its neighbourhood-based replacement operates with a small neighbourhood size and relatively lightweight selection.

Overall, the proposed mechanisms of FairWolf improve search behaviour without introducing prohibitive overhead in practice, and the runtime trend is primarily governed by the diffusion evaluation cost, which is common to all methods.

6.3.3. 2-hop diffusion vs. full IC diffusion

In the main experiments, we adopt a 2-hop diffusion evaluation as a computationally efficient approximation for repeated influence simulations during population-based optimisation. This setting allows all algorithms to be evaluated consistently across multiple networks, propagation probabilities, and independent runs. To further examine whether the use of a finite diffusion horizon affects the main comparative conclusions, we compare the 2-hop setting with full IC diffusion, where the cascade is allowed to propagate until no further activations occur.

We conduct this comparison on the Ham network under three propagation probabilities $p \in \{0.01, 0.05, 0.1\}$. The Ham network is used as a representative benchmark because it has a moderate scale and a non-trivial community structure, making full IC evaluation computationally feasible while still meaningful for assessing community-level diffusion behaviour. All algorithms are evaluated under the same Monte Carlo budget and are repeated 10 times. Table 12 reports the results under full IC. The corresponding 2-hop results are reported in the main comparison tables (Tables 8 to 10), and this section uses full IC as a consistency check to assess whether the 2-hop approximation changes the comparative conclusions.

In summary, the full-IC outcomes exhibit the same performance pattern as the 2-hop setting used in the main study. The relative competitiveness of algorithms remains stable, and the advantages observed for FairWolf are preserved. Under full IC, FairWolf achieves the best hypervolume in all three probability settings, indicating that its non-dominated set maintains strong coverage of the spread-fairness trade-off surface. FairWolf also remains competitive in convergence quality measured by IGD. While the best IGD is obtained by other baselines for $p = 0.01$ and $p = 0.05$, FairWolf stays close to the best values and becomes the best-performing method at $p = 0.1$. The distributional indicators (SP and Δ) show no abnormal degradation for FairWolf under full IC, suggesting that its improved coverage is not achieved by producing irregular or unstable fronts.

6.3.4. Fairness-influence trade-off analysis

Figs. 8 and 9 present the Price of Fairness (POF) and Price of Influence (POI) across all datasets under activation probability $p = 0.01$. FairWolf achieves the lowest POF and POI values on most datasets, ranking second only on *Facebook* and *Wikipedia*, where GFMOGWO performs best. However, FairWolf delivers substantially better spacing and spread on these networks. This indicates that although our algorithm incurs slightly higher trade-off costs in these two cases, it yields a more diverse and evenly distributed set of Pareto solutions, resulting in a more balanced optimisation outcome overall.

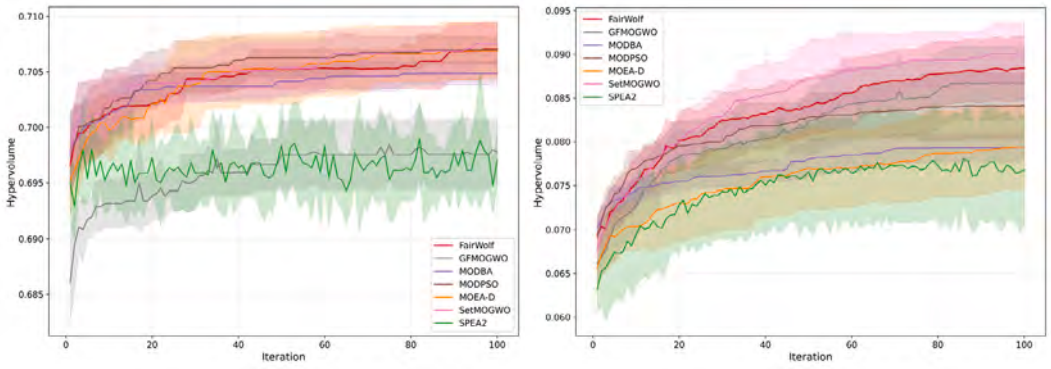
6.3.5. Robustness analysis

In practical settings, diffusion intensity, evaluation approximations, and model parameters may vary. We therefore assess whether the comparative conclusions of FairWolf remain stable under these variations. Across $p \in \{0.01, 0.05, 0.1\}$ and eight networks, FairWolf remains consistently competitive without pronounced performance degradation. The convergence curves further indicate stable optimisation behaviour across runs (Tables 8 to 10 and Fig. 3).

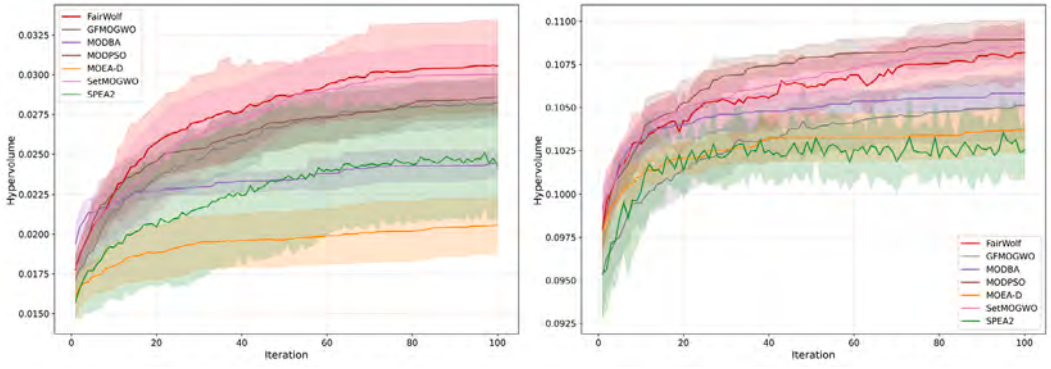
To verify that the 2-hop diffusion approximation does not alter the comparative conclusions, we further compare against a full IC simulation on the Ham network. The full-IC results exhibit the same relative performance pattern and preserve FairWolf's advantages (Section 6.3.3 and Table 12).

Robustness is also supported by sensitivity studies. The default settings for $n_{\text{explorers}}$ and *perturb* provide a strong trade-off across configurations (Table 6), and the fairness-weight analysis (Fair1-Fair3) shows that the joint fairness objective remains effective under reasonable re-weighting, with $\lambda = 0.5$ yielding the most reliable performance.

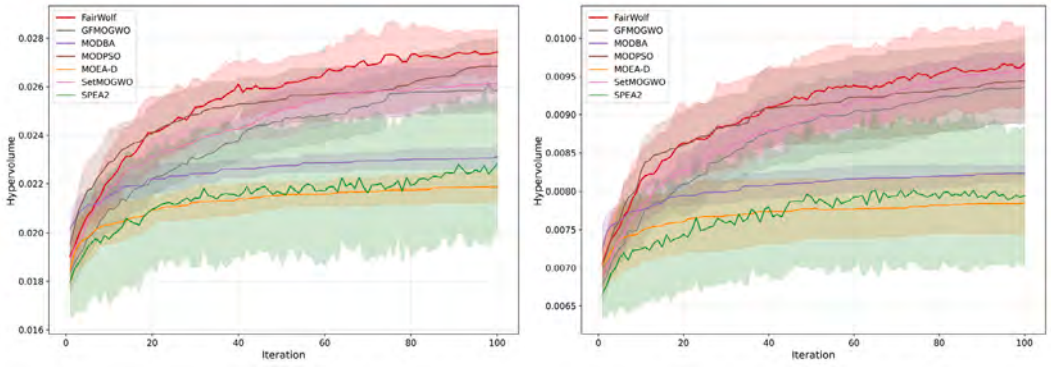
Finally, the per-community fairness analysis on Ham ($p = 0.1$) shows stable worst-case deviations across runs, with no pronounced



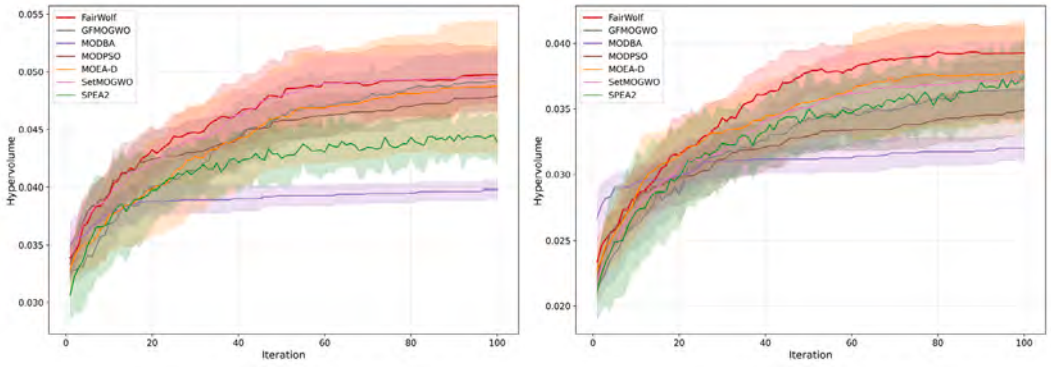
(a) Dolphin (b) Email



(c) Facebook (d) Food

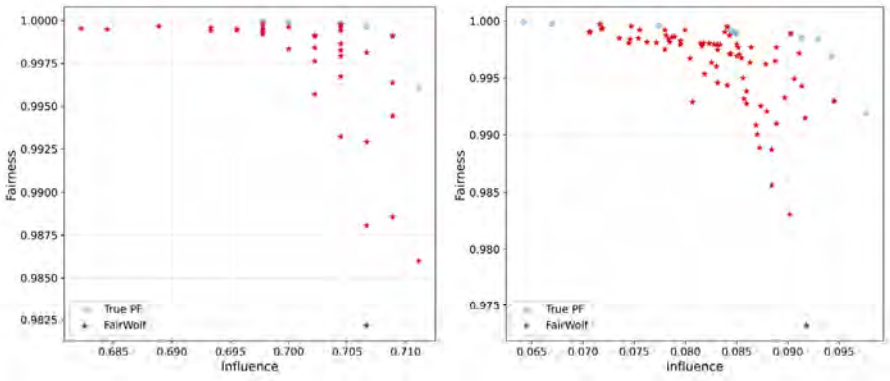


(e) Ham (f) LastFM

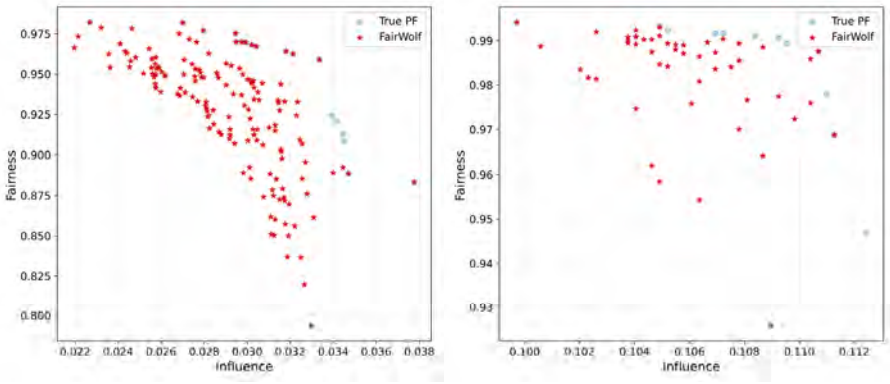


(g) Rice (h) Wiki

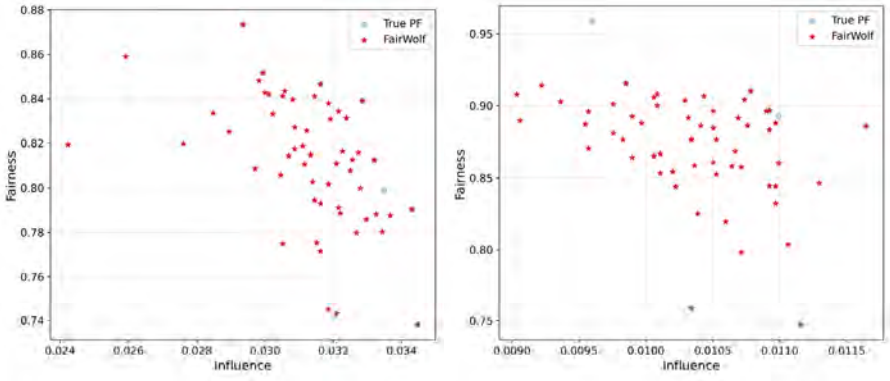
Fig. 3. HV-based convergence curves (mean ± Std) across 8 datasets ($p = 0.01$).



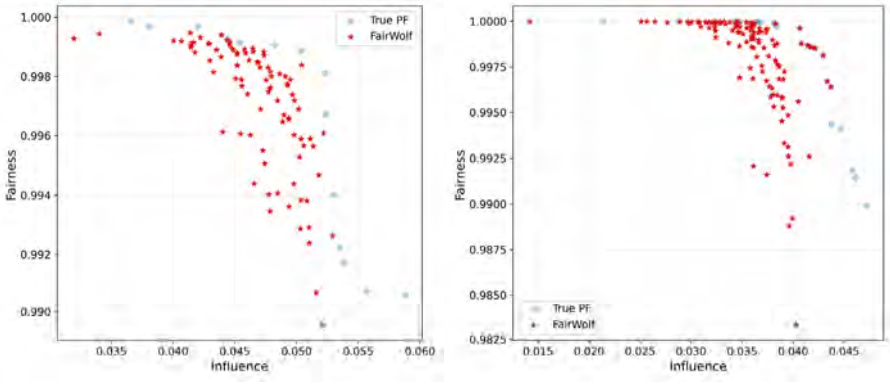
(a) Dolphin (b) Email



(c) Facebook (d) Food

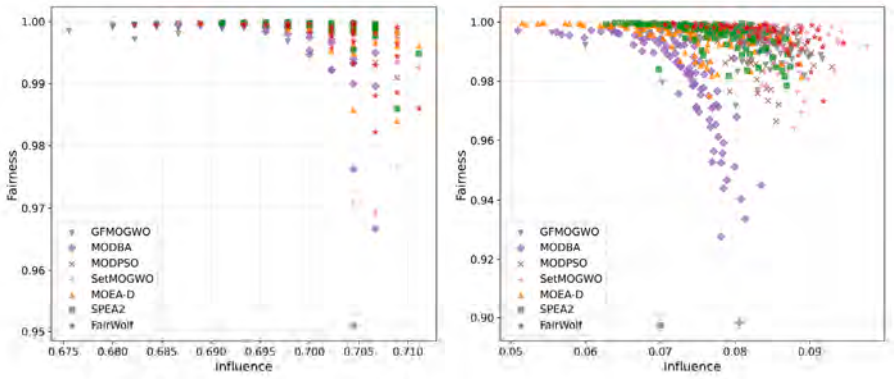


(e) Ham (f) LastFM

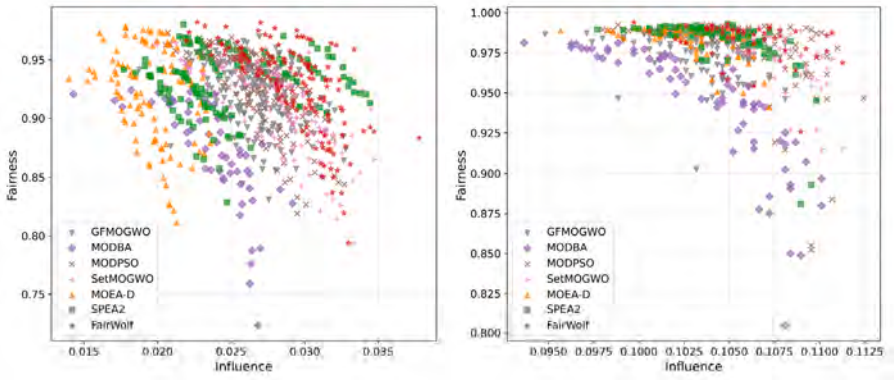


(g) Rice (h) Wiki

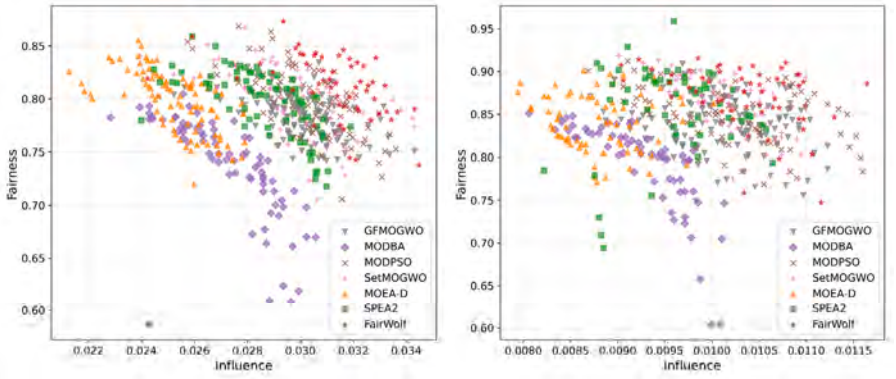
Fig. 4. Comparison between FairWolf solutions and the True PF ($p = 0.01$).



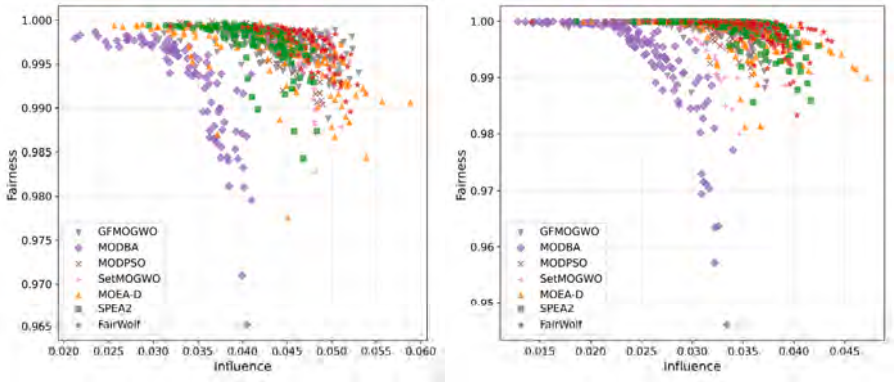
(a) Dolphin (b) Email



(c) Facebook (d) Food



(e) Ham (f) LastFM



(g) Rice (h) Wiki

Fig. 5. Comparison of PF distributions among algorithms ($p = 0.01$).

Table 8
Mean and Std values of evaluation metrics for all algorithms across all networks (k = 30, p = 0.01, Best results in bold).

Datasets	Algorithm	IGD ($\mu \pm \delta$)	HV ($\mu \pm \delta$)	Spacing	Spread	Time ($\mu \pm \delta$)
Dolphin	GFMOGWO	0.0074 ± 0.0022	0.6978 ± 0.0031	**0.0007**	0.0058	0.1069 ± 0.0248
	MODBA	0.0045 ± 0.0010	0.7049 ± 0.0011	0.0030	0.0045	0.0480 ± 0.0082
	MODPSO	**0.0020 ± 0.0004**	0.7070 ± 0.0014	0.0007	0.0028	0.0434 ± 0.0119
	SetMOGWO	0.0030 ± 0.0010	**0.7075 ± 0.0019**	0.0012	0.0028	**0.0410 ± 0.0122**
	MOEA/D	0.0029 ± 0.0012	0.7069 ± 0.0027	0.0011	0.0011	0.0723 ± 0.0094
	SPEA2	0.0022 ± 0.0009	0.6970 ± 0.0023	0.0015	**0.0006**	0.2472 ± 0.0130
	FairWolf	0.0031 ± 0.0008	0.7071 ± 0.0026	0.0014	0.0051	0.0530 ± 0.0153
Facebook	GFMOGWO	0.0138 ± 0.0056	0.0282 ± 0.0015	0.0014	0.0119	1.4486 ± 0.7098
	MODBA	0.0395 ± 0.0063	0.0244 ± 0.0009	0.0056	0.0466	0.8981 ± 0.4552
	MODPSO	0.0106 ± 0.0016	0.0286 ± 0.0014	**0.0012**	0.0244	1.1521 ± 0.5671
	SetMOGWO	0.0151 ± 0.0072	0.0300 ± 0.0020	0.0020	0.0156	1.1694 ± 0.5073
	MOEA/D	0.0200 ± 0.0072	0.0206 ± 0.0018	0.0013	0.0105	0.8856 ± 0.4794
	SPEA2	0.0178 ± 0.0064	0.0243 ± 0.0037	0.0029	0.0161	**0.7919 ± 0.1479**
	FairWolf	**0.0098 ± 0.0044**	**0.0320 ± 0.0015**	0.0025	**0.0000**	1.5464 ± 0.8675
Email	GFMOGWO	0.0058 ± 0.0021	0.0868 ± 0.0045	0.0012	0.0075	0.6133 ± 0.1457
	MODBA	0.0145 ± 0.0015	0.0794 ± 0.0017	0.0032	0.0255	**0.4137 ± 0.1183**
	MODPSO	0.0069 ± 0.0017	0.0841 ± 0.0040	**0.0010**	0.0042	0.4944 ± 0.1675
	SetMOGWO	**0.0054 ± 0.0016**	**0.0900 ± 0.0038**	0.0010	**0.0029**	0.4849 ± 0.1354
	MOEA/D	0.0100 ± 0.0031	0.0794 ± 0.0051	0.0010	0.0118	0.4276 ± 0.1100
	SPEA2	0.0095 ± 0.0049	0.0768 ± 0.0065	0.0083	0.0076	0.7256 ± 0.3364
	FairWolf	0.0061 ± 0.0016	0.0884 ± 0.0038	0.0013	0.0054	0.5451 ± 0.1918
Food	GFMOGWO	0.0067 ± 0.0022	0.1051 ± 0.0021	0.0035	0.0097	0.2114 ± 0.0571
	MODBA	0.0146 ± 0.0009	0.1058 ± 0.0009	0.0052	0.0405	0.1196 ± 0.0309
	MODPSO	**0.0050 ± 0.0014**	**0.1089 ± 0.0012**	0.0031	**0.0009**	0.1143 ± 0.0328
	SetMOGWO	0.0056 ± 0.0013	0.1085 ± 0.0011	**0.0008**	0.0179	**0.1051 ± 0.0295**
	MOEA/D	0.0078 ± 0.0013	0.1037 ± 0.0014	0.0012	0.0055	0.1588 ± 0.0319
	SPEA2	0.0068 ± 0.0016	0.1026 ± 0.0018	0.0027	0.0042	0.3222 ± 0.0366
	FairWolf	0.0067 ± 0.0013	0.1082 ± 0.0017	0.0040	0.0110	0.1278 ± 0.0323
Ham	GFMOGWO	0.0250 ± 0.0074	0.0258 ± 0.0010	**0.0008**	0.0259	0.3536 ± 0.0219
	MODBA	0.0412 ± 0.0054	0.0231 ± 0.0005	0.0016	0.0686	0.2497 ± 0.0686
	MODPSO	**0.0166 ± 0.0080**	0.0268 ± 0.0012	0.0027	0.0099	0.2376 ± 0.0167
	SetMOGWO	0.0179 ± 0.0050	0.0261 ± 0.0011	0.0026	0.0211	0.2126 ± 0.0194
	MOEA/D	0.0181 ± 0.0052	0.0219 ± 0.0006	0.0025	0.0205	**0.2113 ± 0.0370**
	SPEA2	0.0228 ± 0.0122	0.0228 ± 0.0028	0.0148	0.0254	0.4218 ± 0.0468
	FairWolf	0.0178 ± 0.0076	**0.0274 ± 0.0010**	0.0022	**0.0000**	0.2956 ± 0.0532
LastFM	GFMOGWO	0.0334 ± 0.0188	0.0094 ± 0.0005	0.0015	0.0613	0.3557 ± 0.0710
	MODBA	0.0707 ± 0.0070	0.0082 ± 0.0001	0.0059	0.1237	0.3144 ± 0.0639
	MODPSO	0.0238 ± 0.0116	0.0094 ± 0.0006	0.0016	0.0745	0.1923 ± 0.0440
	SetMOGWO	0.0177 ± 0.0062	0.0096 ± 0.0002	0.0011	0.0237	0.1771 ± 0.0350
	MOEA/D	0.0327 ± 0.0103	0.0078 ± 0.0004	**0.0011**	0.0404	**0.1257 ± 0.0105**
	SPEA2	0.0435 ± 0.0522	0.0079 ± 0.0010	0.0065	0.0461	0.3177 ± 0.0339
	FairWolf	**0.0164 ± 0.0088**	**0.0097 ± 0.0005**	0.0024	**0.0217**	0.2165 ± 0.0426
Wiki	GFMOGWO	0.0048 ± 0.0016	0.0365 ± 0.0023	0.0004	0.0083	1.7236 ± 0.4642
	MODBA	0.0118 ± 0.0010	0.0320 ± 0.0010	0.0012	0.0121	**1.1060 ± 0.3770**
	MODPSO	0.0062 ± 0.0014	0.0349 ± 0.0019	**0.0004**	0.0064	1.3835 ± 0.4368
	SetMOGWO	0.0052 ± 0.0021	0.0371 ± 0.0031	0.0006	0.0069	1.4626 ± 0.4890
	MOEA/D	0.0045 ± 0.0021	0.0378 ± 0.0041	0.0005	**0.0055**	1.1086 ± 0.1966
	SPEA2	0.0039 ± 0.0017	0.0375 ± 0.0029	0.0007	0.0064	1.2045 ± 0.1407
	FairWolf	**0.0036 ± 0.0010**	**0.0393 ± 0.0022**	0.0011	0.0058	1.6706 ± 0.5842
Rice	GFMOGWO	0.0037 ± 0.0013	0.0491 ± 0.0028	0.0004	0.0051	4.1600 ± 0.2454
	MODBA	0.0122 ± 0.0012	0.0398 ± 0.0009	0.0009	0.0171	2.8204 ± 0.1978
	MODPSO	0.0042 ± 0.0008	0.0479 ± 0.0013	**0.0003**	0.0047	3.5178 ± 0.2907
	SetMOGWO	**0.0035 ± 0.0009**	0.0495 ± 0.0021	0.0006	0.0041	3.5969 ± 0.2002
	MOEA/D	0.0045 ± 0.0034	0.0488 ± 0.0060	0.0012	**0.0027**	**2.0287 ± 0.6590**
	SPEA2	0.0051 ± 0.0014	0.0439 ± 0.0020	0.0005	0.0062	2.4615 ± 0.6019
	FairWolf	0.0036 ± 0.0012	**0.0497 ± 0.0026**	0.0004	0.0055	4.0919 ± 0.4233

systematic over-/under-allocation concentrated in a small subset of communities (Figs. 6 to 7).

Overall, these results suggest that FairWolf exhibits robust performance and fairness behaviour under the main sources of variation considered in this study.

7. Conclusion and future work

In this paper, we formulated the Fair Multi-objective Influence Maximisation (FMOIM) problem, which extends classical influence maximisation by jointly optimising influence spread and fairness. Instead of using a single fairness proxy, we model equity fairness at the community level as the alignment between the realised diffusion-benefit distribution and a desired reference allocation. Jensen–Shannon divergence (JSD) captures distributional deviation from the reference, while Jain's fairness index characterises the evenness of benefit allocation among communities. This formulation provides an explicit and measurable way to study the spread-fairness trade-off under a fixed seed budget.

To address FMOIM, we propose FairWolf, a problem-driven discrete multi-objective optimisation model that reformulates Grey Wolf Optimiser dynamics to search directly over fixed-budget seed sets under community-level fairness objectives. FairWolf is designed around the seed-set decision space and the fairness-aware objective structure, so that the search naturally produces diverse Pareto-optimal seed sets that balance diffusion effectiveness and equitable allocation. It combines

Table 9
Mean and Std values of evaluation metrics for all algorithms across all networks (k = 30, p = 0.05, Best results in bold).

Datasets	Algorithm	IGD ($\mu \pm \delta$)	HV ($\mu \pm \delta$)	Spacing	Spread	Time ($\mu \pm \delta$)
Dolphin	GFMOGWO	0.0060 ± 0.0030	0.7766 ± 0.0075	0.0015	0.0045	0.1042 ± 0.0111
	MODBA	0.0051 ± 0.0021	0.7870 ± 0.0044	0.0017	0.0048	0.0490 ± 0.0020
	MODPSO	0.0068 ± 0.0031	0.7878 ± 0.0037	**0.0004**	0.0033	0.0450 ± 0.0046
	SetMOGWO	0.0087 ± 0.0052	0.7886 ± 0.0061	0.0013	0.0122	**0.0395 ± 0.0030**
	MOEA/D	0.0067 ± 0.0036	0.7884 ± 0.0027	0.0012	0.0041	0.0809 ± 0.0120
	SPEA2	**0.0048 ± 0.0039**	0.7710 ± 0.0066	0.0006	**0.0020**	0.2544 ± 0.0120
	FairWolf	0.0172 ± 0.0124	**0.7895 ± 0.0046**	0.0017	0.0200	0.0507 ± 0.0040
Facebook	GFMOGWO	0.0291 ± 0.0091	0.1991 ± 0.0168	0.0028	0.0322	4.9387 ± 1.1279
	MODBA	0.0659 ± 0.0052	0.1707 ± 0.0063	0.0032	0.0599	2.8622 ± 0.5301
	MODPSO	0.0328 ± 0.0091	0.1894 ± 0.0110	0.0027	0.0340	3.6661 ± 0.8664
	SetMOGWO	0.0335 ± 0.0074	0.1922 ± 0.0094	0.0022	0.0498	3.4332 ± 0.7439
	MOEA/D	0.0398 ± 0.0191	0.1798 ± 0.0186	0.0026	0.0244	**1.4808 ± 0.3417**
	SPEA2	**0.0176 ± 0.0095**	**0.1992 ± 0.0165**	**0.0014**	**0.0160**	1.9048 ± 0.1836
	FairWolf	0.0320 ± 0.0106	0.1921 ± 0.0167	0.0021	0.0206	4.0039 ± 0.8709
Email	GFMOGWO	0.0182 ± 0.0062	0.3942 ± 0.0116	0.0012	0.0321	1.9594 ± 0.0832
	MODBA	0.0442 ± 0.0059	0.3590 ± 0.0073	0.0040	0.0588	1.3369 ± 0.0565
	MODPSO	0.0184 ± 0.0063	0.3948 ± 0.0124	**0.0011**	0.0151	1.6980 ± 0.1434
	SetMOGWO	0.0156 ± 0.0061	0.3999 ± 0.0122	0.0012	0.0100	1.6077 ± 0.0766
	MOEA/D	**0.0088 ± 0.0035**	**0.4153 ± 0.0099**	0.0019	**0.0081**	**0.8397 ± 0.0562**
	SPEA2	0.0210 ± 0.0099	0.3820 ± 0.0184	0.0012	0.0319	0.9292 ± 0.0505
	FairWolf	0.0163 ± 0.0048	0.4040 ± 0.0112	0.0017	0.0443	1.7774 ± 0.1180
Food	GFMOGWO	0.0089 ± 0.0027	0.2044 ± 0.0061	0.0019	0.0139	0.4007 ± 0.0570
	MODBA	0.0218 ± 0.0027	0.1975 ± 0.0027	0.0043	0.0327	0.2485 ± 0.0235
	MODPSO	0.0075 ± 0.0029	0.2076 ± 0.0057	0.0021	**0.0000**	0.2500 ± 0.0457
	SetMOGWO	**0.0074 ± 0.0026**	0.2067 ± 0.0054	0.0017	0.0058	0.2316 ± 0.0209
	MOEA/D	0.0102 ± 0.0028	0.2028 ± 0.0046	0.0019	0.0097	**0.1700 ± 0.0121**
	SPEA2	0.0107 ± 0.0033	0.1919 ± 0.0043	0.0025	0.0051	0.3248 ± 0.0194
	FairWolf	0.0081 ± 0.0019	**0.2085 ± 0.0037**	**0.0013**	0.0209	0.2695 ± 0.0319
Ham	GFMOGWO	0.0217 ± 0.0089	0.1272 ± 0.0059	0.0018	0.0164	1.2503 ± 0.1861
	MODBA	0.0656 ± 0.0047	0.1023 ± 0.0030	0.0029	0.0815	0.7238 ± 0.0939
	MODPSO	0.0295 ± 0.0099	0.1201 ± 0.0071	0.0024	0.0397	0.9484 ± 0.1394
	SetMOGWO	0.0218 ± 0.0072	0.1261 ± 0.0067	0.0020	0.0127	0.8989 ± 0.1160
	MOEA/D	0.0310 ± 0.0081	0.1141 ± 0.0093	**0.0020**	0.0210	**0.3896 ± 0.0475**
	SPEA2	0.0232 ± 0.0121	0.1219 ± 0.0072	0.0029	0.0179	0.6187 ± 0.0611
	FairWolf	**0.0191 ± 0.0094**	**0.1275 ± 0.0071**	0.0025	**0.0000**	1.0056 ± 0.1847
LastFM	GFMOGWO	0.0147 ± 0.0048	0.0372 ± 0.0019	0.0011	0.0180	0.5361 ± 0.0551
	MODBA	0.0291 ± 0.0023	0.0278 ± 0.0012	0.0031	0.0308	0.3671 ± 0.0234
	MODPSO	0.0119 ± 0.0029	0.0346 ± 0.0026	0.0021	**0.0048**	0.3340 ± 0.0350
	SetMOGWO	0.0113 ± 0.0034	0.0360 ± 0.0024	**0.0011**	0.0198	0.3286 ± 0.0493
	MOEA/D	0.0169 ± 0.0039	0.0266 ± 0.0036	0.0015	0.0225	**0.2083 ± 0.0241**
	SPEA2	0.0219 ± 0.0188	0.0304 ± 0.0086	0.0048	0.0084	0.4226 ± 0.0788
	FairWolf	**0.0091 ± 0.0018**	**0.0384 ± 0.0023**	0.0023	0.0069	0.3994 ± 0.0399
Wiki	GFMOGWO	0.0202 ± 0.0028	0.2761 ± 0.0111	0.0007	0.0641	7.2760 ± 0.4126
	MODBA	0.0323 ± 0.0027	0.2439 ± 0.0049	0.0024	0.0507	4.6227 ± 0.2612
	MODPSO	0.0207 ± 0.0051	0.2642 ± 0.0123	0.0008	0.0447	5.7122 ± 0.5467
	SetMOGWO	0.0159 ± 0.0031	0.2727 ± 0.0116	0.0008	**0.0256**	5.4532 ± 0.4263
	MOEA/D	0.0158 ± 0.0078	0.3273 ± 0.0114	0.0036	0.0286	3.8689 ± 0.1936
	SPEA2	**0.0137 ± 0.0061**	**0.2914 ± 0.0153**	0.0007	0.0347	**3.0281 ± 0.2867**
	FairWolf	0.0178 ± 0.0043	0.2743 ± 0.0116	**0.0006**	0.0411	6.5359 ± 0.6674
Rice	GFMOGWO	0.0307 ± 0.0078	0.4094 ± 0.0172	0.0012	0.0276	10.9338 ± 1.7995
	MODBA	0.0672 ± 0.0029	0.3523 ± 0.0037	0.0025	0.0811	**6.8409 ± 1.9835**
	MODPSO	0.0294 ± 0.0048	0.4095 ± 0.0095	0.0050	0.0560	9.7098 ± 1.8284
	SetMOGWO	0.0370 ± 0.0077	0.3967 ± 0.0123	**0.0009**	0.0377	9.4331 ± 0.6581
	MOEA/D	0.0350 ± 0.0142	**0.4669 ± 0.0136**	0.0047	**0.0082**	11.2371 ± 0.7543
	SPEA2	**0.0278 ± 0.0088**	0.4347 ± 0.0197	0.0025	0.0195	9.7565 ± 0.8766
	FairWolf	0.0318 ± 0.0073	0.4210 ± 0.0189	0.0013	0.0262	10.8342 ± 1.7873

Table 10
Mean and Std values of evaluation metrics for all algorithms across all networks (k = 30, p = 0.1, Best results in bold).

Datasets	Algorithm	IGD ($\mu \pm \delta$)	HV ($\mu \pm \delta$)	Spacing	Spread	Time ($\mu \pm \delta$)
Dolphin	GFMOGWO	0.0114 ± 0.0051	0.8506 ± 0.0088	**0.0009**	0.0222	0.1904 ± 0.0337
	MODBA	0.0092 ± 0.0054	0.8564 ± 0.0038	0.0021	0.0202	0.0722 ± 0.0206
	MODPSO	0.0074 ± 0.0059	0.8593 ± 0.0054	0.0023	0.0034	0.0652 ± 0.0194
	SetMOGWO	0.0096 ± 0.0056	0.8646 ± 0.0069	0.0015	**0.0005**	0.0692 ± 0.0189
	MOEA/D	**0.0073 ± 0.0057**	0.8655 ± 0.0037	0.0010	0.0034	0.1524 ± 0.0365
	SPEA2	0.0102 ± 0.0084	0.8460 ± 0.0089	0.0018	0.0008	0.4975 ± 0.0163
	FairWolf	0.0212 ± 0.0157	**0.8675 ± 0.0060**	0.0028	0.0144	**0.0552 ± 0.0033**
Facebook	GFMOGWO	0.0414 ± 0.0132	0.4014 ± 0.0215	0.0039	0.0266	3.2575 ± 0.2350
	MODBA	0.0778 ± 0.0070	0.3720 ± 0.0127	0.0037	0.0975	4.2882 ± 1.3094
	MODPSO	0.0429 ± 0.0164	0.3913 ± 0.0256	0.0021	0.0295	5.1868 ± 1.1519
	SetMOGWO	0.0452 ± 0.0135	0.3909 ± 0.0179	0.0017	0.0502	4.4347 ± 1.4532
	MOEA/D	0.0369 ± 0.0168	0.4018 ± 0.0330	0.0023	0.0207	**2.8791 ± 0.7589**
	SPEA2	**0.0291 ± 0.0119**	0.4146 ± 0.0289	0.0023	**0.0000**	3.2447 ± 1.0529
	FairWolf	0.0332 ± 0.0115	**0.4187 ± 0.0160**	**0.0017**	0.0367	3.2152 ± 0.5080
Email	GFMOGWO	0.0195 ± 0.0067	0.6638 ± 0.0093	0.0006	0.0430	2.0963 ± 0.4540
	MODBA	0.0509 ± 0.0050	0.6261 ± 0.0051	0.0049	0.0536	1.2165 ± 0.2041
	MODPSO	0.0232 ± 0.0068	0.6584 ± 0.0079	0.0023	0.0184	1.3926 ± 0.2126
	SetMOGWO	0.0214 ± 0.0077	0.6633 ± 0.0110	0.0020	0.0348	1.3406 ± 0.2289
	MOEA/D	**0.0109 ± 0.0027**	**0.6841 ± 0.0073**	**0.0004**	0.0412	1.1230 ± 0.0603
	SPEA2	0.0299 ± 0.0250	0.6380 ± 0.0532	0.0016	**0.0081**	**1.1162 ± 0.1380**
	FairWolf	0.0192 ± 0.0090	0.6682 ± 0.0128	0.0025	0.0327	1.9964 ± 0.2346
Food	GFMOGWO	0.0120 ± 0.0015	0.3353 ± 0.0063	0.0011	0.0291	0.3880 ± 0.0982
	MODBA	0.0252 ± 0.0018	0.3208 ± 0.0040	0.0018	0.0409	0.2782 ± 0.0396
	MODPSO	0.0103 ± 0.0034	0.3401 ± 0.0112	0.0017	0.0077	0.2712 ± 0.0503
	SetMOGWO	0.0114 ± 0.0035	0.3379 ± 0.0087	0.0016	0.0109	0.2543 ± 0.0480
	MOEA/D	**0.0098 ± 0.0018**	0.3419 ± 0.0048	0.0012	0.0111	0.2479 ± 0.0527
	SPEA2	0.0120 ± 0.0040	0.3271 ± 0.0103	**0.0007**	**0.0068**	0.4441 ± 0.1054
	FairWolf	0.0153 ± 0.0054	**0.3422 ± 0.0054**	0.0012	0.0274	**0.2066 ± 0.0507**
Ham	GFMOGWO	**0.0196 ± 0.0071**	**0.2907 ± 0.0118**	**0.0015**	0.0535	1.0517 ± 0.1724
	MODBA	0.0967 ± 0.0083	0.2290 ± 0.0027	0.0031	0.0699	1.0130 ± 0.2137
	MODPSO	0.0329 ± 0.0069	0.2770 ± 0.0072	0.0019	**0.0138**	1.4335 ± 0.2347
	SetMOGWO	0.0229 ± 0.0074	0.2880 ± 0.0099	0.0025	0.0450	1.3476 ± 0.2570
	MOEA/D	0.0331 ± 0.0107	0.2775 ± 0.0125	0.0020	0.0332	**0.6863 ± 0.0397**
	SPEA2	0.0203 ± 0.0110	0.2889 ± 0.0129	0.0023	0.0388	0.8903 ± 0.0745
	FairWolf	0.0211 ± 0.0063	0.2888 ± 0.0089	0.0024	0.0578	0.9254 ± 0.1711
LastFM	GFMOGWO	0.0120 ± 0.0026	0.0869 ± 0.0026	0.0018	0.0214	0.9212 ± 0.2702
	MODBA	0.0353 ± 0.0013	0.0660 ± 0.0030	0.0024	0.0287	0.6879 ± 0.1647
	MODPSO	0.0143 ± 0.0027	0.0833 ± 0.0041	**0.0013**	0.0152	0.7625 ± 0.1791
	SetMOGWO	0.0147 ± 0.0048	0.0828 ± 0.0072	0.0019	0.0173	0.6779 ± 0.2244
	MOEA/D	0.0183 ± 0.0084	0.0772 ± 0.0116	0.0014	0.0154	**0.3828 ± 0.0412**
	SPEA2	0.0212 ± 0.0186	0.0762 ± 0.0216	0.0031	**0.0110**	0.6094 ± 0.1203
	FairWolf	**0.0104 ± 0.0026**	**0.0881 ± 0.0036**	0.0020	0.0133	0.7913 ± 0.1845
Wiki	GFMOGWO	0.1357 ± 0.0137	0.5424 ± 0.0148	**0.0005**	0.1706	12.3681 ± 1.0281
	MODBA	0.0563 ± 0.0103	0.4810 ± 0.0067	0.0034	0.0579	**6.2818 ± 1.7139**
	MODPSO	0.0623 ± 0.0401	0.5240 ± 0.0196	0.0019	0.0426	8.4204 ± 2.6604
	SetMOGWO	**0.0533 ± 0.0386**	0.5183 ± 0.0204	0.0012	**0.0182**	7.4363 ± 2.1583
	MOEA/D	0.1592 ± 0.0057	**0.5846 ± 0.0066**	0.0005	0.1716	8.5593 ± 2.4783
	SPEA2	0.1203 ± 0.0530	0.5573 ± 0.0245	0.0111	0.0278	6.9919 ± 2.6848
	FairWolf	0.1022 ± 0.0549	0.5400 ± 0.0257	0.0020	0.0312	10.8054 ± 1.7316
Rice	GFMOGWO	0.0169 ± 0.0076	0.7442 ± 0.0084	0.0031	0.0053	14.2354 ± 4.0043
	MODBA	0.0944 ± 0.0054	0.6658 ± 0.0054	0.0012	0.1552	**10.0876 ± 2.6077**
	MODPSO	0.0396 ± 0.0129	0.7209 ± 0.0129	0.0008	0.0394	13.4392 ± 4.4322
	SetMOGWO	0.0335 ± 0.0112	0.7270 ± 0.0112	0.0005	0.0330	13.8599 ± 3.4352
	MOEA/D	**0.0047 ± 0.0009**	**0.7635 ± 0.0040**	**0.0002**	**0.0019**	12.4462 ± 3.6568
	SPEA2	0.0330 ± 0.0378	0.7221 ± 0.0453	0.0033	0.0244	10.2405 ± 4.1268
	FairWolf	0.0183 ± 0.0047	0.7426 ± 0.0059	0.0035	0.0270	14.4553 ± 3.4396

Table 11
Average ranking of algorithms across datasets and Pareto-front quality indicators under different p values. A smaller rank indicates better overall performance.

Algorithm	$p = 0.01$	$p = 0.05$	$p = 0.1$	Overall
GFMOGWO	3.94	3.59	3.66	3.73
MODBA	6.25	6.41	6.31	6.32
MODPSO	3.00	3.97	3.84	3.60
SetMOGWO	2.97	3.34	3.88	3.40
MOEA/D	3.88	3.88	**2.91**	3.55
SPEA2	5.00	3.59	3.75	4.11
FairWolf	**2.88**	**3.22**	3.63	**3.24**

Table 12
Ham network: full IC results under $p \in \{0.01, 0.05, 0.1\}$.

Full IC	Algorithm	IGD ($\mu \pm \delta$)	HV ($\mu \pm \delta$)	Spacing	Spread	Time ($\mu \pm \delta$)
p = 0.01	GFMOGWO	0.0157 ± 0.0073	0.0285 ± 0.0014	0.0013	0.0177	0.3151 ± 0.0619
	MODBA	0.0394 ± 0.0070	0.0255 ± 0.0007	0.0029	0.0560	0.2048 ± 0.0190
	MODPSO	0.0137 ± 0.0046	0.0292 ± 0.0017	0.0015	**0.0078**	0.1894 ± 0.0197
	SetMOGWO	**0.0130 ± 0.0045**	0.0292 ± 0.0010	0.0014	0.0185	**0.1872 ± 0.0301**
	MOEA-D	0.0186 ± 0.0036	0.0248 ± 0.0010	0.0011	0.0225	0.2133 ± 0.0573
	SPEA2	0.0197 ± 0.0093	0.0240 ± 0.0019	**0.0010**	0.0231	0.4239 ± 0.0830
	FairWolf	0.0167 ± 0.0048	**0.0294 ± 0.0011**	0.0041	0.0160	0.2461 ± 0.0707
p = 0.05	GFMOGWO	0.0162 ± 0.0061	0.3317 ± 0.0036	0.0023	0.0191	2.1163 ± 0.5303
	MODBA	0.0501 ± 0.0049	0.3203 ± 0.0020	**0.0015**	0.0829	1.7817 ± 0.2418
	MODPSO	**0.0116 ± 0.0037**	0.3370 ± 0.0023	0.0026	0.0390	**1.7353 ± 0.3746**
	SetMOGWO	0.0160 ± 0.0070	0.3350 ± 0.0030	0.0017	0.0194	1.7977 ± 0.4446
	MOEA-D	0.0162 ± 0.0077	0.3332 ± 0.0050	0.0031	**0.0090**	2.0073 ± 0.5354
	SPEA2	0.0186 ± 0.0169	0.3286 ± 0.0097	0.0026	0.0107	1.9789 ± 0.2944
	FairWolf	0.0135 ± 0.0080	**0.3372 ± 0.0030**	0.0053	0.0220	1.9469 ± 0.2876
p = 0.1	GFMOGWO	0.0089 ± 0.0052	0.5817 ± 0.0030	0.0012	0.0145	3.8505 ± 1.4270
	MODBA	0.0314 ± 0.0027	0.5664 ± 0.0017	0.0013	0.0416	3.1626 ± 1.1767
	MODPSO	0.0110 ± 0.0049	0.5838 ± 0.0048	0.0023	**0.0000**	**2.9963 ± 1.2633**
	SetMOGWO	0.0087 ± 0.0041	0.5845 ± 0.0035	0.0030	0.0092	3.1327 ± 1.3657
	MOEA-D	0.0120 ± 0.0054	0.5795 ± 0.0045	**0.0012**	0.0082	3.4080 ± 1.4020
	SPEA2	0.0152 ± 0.0098	0.5721 ± 0.0111	0.0020	0.0192	3.8896 ± 1.5333
	FairWolf	**0.0069 ± 0.0035**	**0.5849 ± 0.0033**	0.0026	0.0059	3.8211 ± 1.3456

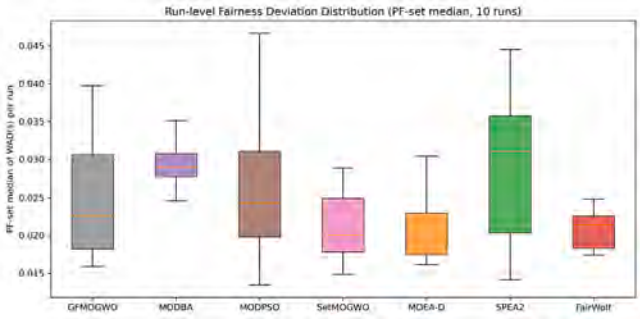


Fig. 6. Ham network ($p = 0.1$): run-level distribution of PF-set median $WAD(s)$ over 10 runs.

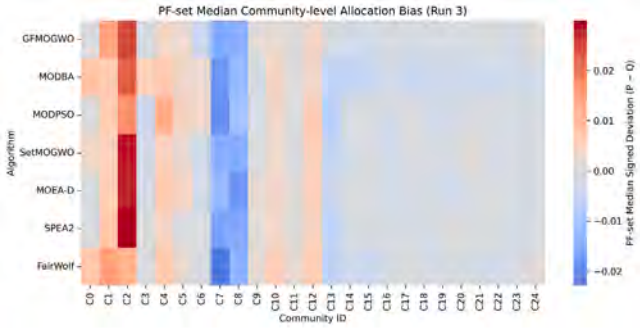


Fig. 7. Ham network ($p = 0.1$): community-level signed allocation deviation δ_j for the PF-set median solution in run 3.

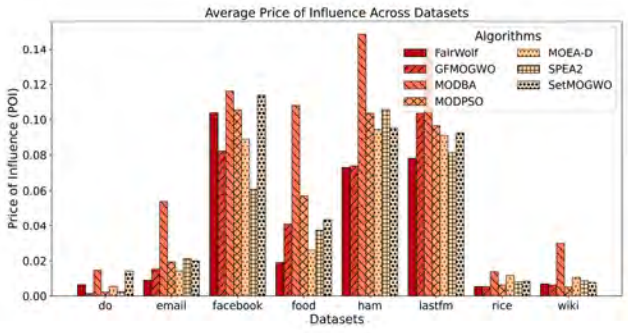


Fig. 8. Average Price of Influence (POI) obtained by different algorithms across 8 datasets ($p = 0.01$).

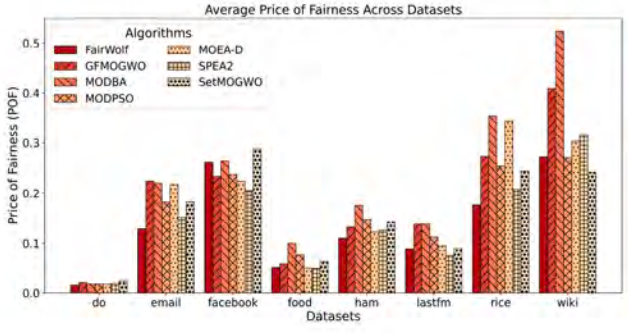


Fig. 9. Average Price of Fairness (POF) obtained by different algorithms across 8 datasets ($p = 0.01$).

discrete position updating, an Explorer-Augmented Leader Selection strategy to encourage coverage of underexplored regions, and an HV-triggered perturbation mechanism to reduce stagnation in non-convex multi-objective landscapes.

Extensive experiments on eight real-world networks across various propagation probabilities demonstrate that FairWolf achieves competitive performance across multiple quality metrics. It yields higher hypervolume, competitive IGD, and well-distributed Pareto fronts in terms of Spread and Spacing, while maintaining practical computational cost compared with more complex swarm-based baselines. It delivers higher hypervolume, competitive IGD, and well-distributed Pareto fronts in terms of Spread and Spacing, while maintaining a lower computational cost than more complex swarm-based baselines. Beyond these empirical gains, the resulting Pareto fronts have direct practical value. They provide practitioners with a set of candidate seed sets that trade off spread and fairness, enabling the selection of an operating point aligned with domain priorities (e.g., equitable access in public-health messaging versus maximum reach under strict budgets). The fairness

objectives further offer transparent diagnostics of community-level allocation, helping identify persistent under-service and supporting more accountable outreach decisions in real deployments.

Future work may extend this study in two directions. The first is to adapt FairWolf to dynamic networks in which diffusion patterns evolve over time. The second is to apply the proposed mechanisms to other multi-objective optimisation problems beyond influence maximisation, such as resource allocation or recommendation, where balancing efficiency and fairness remains a central challenge.

CRediT authorship contribution statement

Ziying Zhao: Writing – review & editing, Writing – original draft, Software, Methodology. **Weihua Li:** Writing – review & editing, Validation, Supervision. **Jing Ma:** Supervision. **Jianhua Jiang:** Writing – review & editing, Supervision. **Quan Bai:** Writing – review & editing. **Wen Gu:** Writing – review & editing.

Declaration of competing interest

The authors declared that they have no conflicts of interest to this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Data availability

Data will be made available on request.

References

[1] K. Aggarwal, A. Arora, Influence maximisation in social networks using discrete BAT-modified (DBATM) optimisation algorithm: A computationally intelligent viral marketing approach, Soc. Netw. Anal. Min. 13 (1) (2023) 146.

[2] A. Yadav, B. Wilder, E. Rice, R. Petering, J. Craddock, A. Yoshioka-Maxwell, M. Hemler, L. Onasch-Vera, M. Tambe, D. Woo, Bridging the gap between theory and practice in influence maximisation: Raising awareness about HIV among homeless youth, in: Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018, pp. 5399–5403.

[3] O. Magdaci, Y. Matalon, D. Yamin, Modeling the debate dynamics of political communication in social media networks, Expert Syst. Appl. 206 (2022) 117782.

[4] M. Ye, X. Liu, W.-C. Lee, Exploring social influence for recommendation: a generative model approach, in: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'12, Association for Computing Machinery, New York, NY, USA, 2012, pp. 671–680.

[5] D. Kempe, J. Kleinberg, É. Tardos, Maximising the spread of influence through a social network, in: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003, pp. 137–146.

[6] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, N. Glance, Cost-effective outbreak detection in networks, in: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2007, pp. 420–429.

[7] L.C. Freeman, Centrality in social networks conceptual clarification, Soc. Netw. 1 (3) (1978) 215–239.

[8] L.C. Freeman, A set of measures of centrality based on betweenness, Sociometry 40 (1) (1977) 35–41.

[9] J. Ali, M. Babaei, A. Chakraborty, B. Mirzasoleiman, P.K. Gummadi, A. Singla, On the fairness of time-critical influence maximisation in social networks (extended abstract), in: Proceedings of the IEEE International Conference on Data Engineering, ICDE, 2022, pp. 1541–1542.

[10] A.-A. Stoica, A. Chaintreau, Fairness in social influence maximisation, in: Proceedings of the World Wide Web Conference Companion, 2019, pp. 569–574.

[11] A. Tsang, B. Wilder, E. Rice, M. Tambe, Y. Zick, Group-fairness in influence maximisation, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI, 2019, pp. 5997–6005.

[12] G. Farnad, B. Babaki, M. Gendreau, A unifying framework for fairness-aware influence maximisation, in: Proceedings of the Web Conference Companion, 2020, pp. 714–722.

[13] A. Rahmattalabi, S. Jabbari, H. Lakkaraju, P. Vayanos, M. Izenberg, R. Brown, E. Rice, M. Tambe, Fair influence maximization: A welfare optimization approach, in: Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI, 2021, pp. 11630–11638.

[14] J. Lin, Divergence measures based on the Shannon entropy, IEEE Trans. Inf. Theory 37 (1) (1991) 145–151.

[15] R. Jain, D.M. Chiu, W.R. Hawe, A quantitative measure of fairness and discrimination for resource allocation in shared computer systems, Comput. Res. Repos. (CoRR) cs.NI/9809099 (1998).

[16] C. Borgs, M. Brautbar, J. Chayes, B. Lucier, Maximizing social influence in nearly optimal time, in: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2014, pp. 946–957.

[17] J. Tang, R. Zhang, P. Wang, Z. Zhao, L. Fan, X. Liu, A discrete shuffled frog-leaping algorithm to identify influential nodes for influence maximization in social networks, Knowl.-Based Syst. 187 (2020) 104833.

[18] C. Salavati, A. Abdollahpouri, Identifying influential nodes based on ant colony optimization to maximize profit in social networks, Swarm Evol. Comput. 51 (2019) 100614.

[19] L. Zhang, Y. Liu, F. Cheng, J. Qiu, X. Zhang, A local-global influence indicator based constrained evolutionary algorithm for budgeted influence maximization in social networks, IEEE Trans. Netw. Sci. Eng. 8 (2) (2021) 1557–1570.

[20] L. Ma, Z. Shao, X. Li, Q. Lin, J. Li, V.C.M. Leung, A.K. Nandi, Influence maximization in complex networks by using evolutionary deep reinforcement learning, IEEE Trans. Emerg. Top. Comput. Intell. 7 (4) (2023) 995–1009.

[21] S. Tian, S. Mo, L. Wang, Z. Peng, Deep reinforcement learning-based approach to tackle topic-aware influence maximization, Data Sci. Eng. 5 (1) (2020) 1–11.

[22] B. Fish, A. Bashardoust, D. Boyd, S. Friedler, C. Scheidegger, S. Venkatasubramanian, Gaps in information access in social networks? in: Proceedings of the World Wide Web Conference, 2019, pp. 480–490.

[23] A.-A. Stoica, J.X. Han, A. Chaintreau, Seeding network influence in biased networks and the benefits of diversity, in: Proceedings of the Web Conference 2020, WWW'20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 2089–2098.

[24] B. Razaghi, M. Roayaei, M.N. Charkari, On the group-fairness-aware influence maximisation in social networks, IEEE Trans. Comput. Soc. Syst. 10 (6) (2023) 3406–3414.

[25] H. Gong, C. Guo, Influence maximisation considering fairness: A multi-objective optimisation approach with prior knowledge, Expert Syst. Appl. 214 (2023) 119138.

[26] K. Ma, X. Xu, H. Yang, R. Cao, L. Zhang, Fair influence maximisation in social networks: A community-based evolutionary algorithm, IEEE Trans. Emerg. Top. Comput. (2024) 1–13.

[27] Z. Zhao, W. Li, J. Ma, J. Jiang, Q. Bai, Fair influence maximisation in social networks: A group-fairness-aware multi-objective grey wolf optimiser, in: Proceedings of the IEEE International Conference on Agents, ICA, 2024, pp. 88–93.

[28] J.-R. Lee, C.-W. Chung, A fast approximation for influence maximisation in large social networks, in: Proceedings of the 23rd International Conference on World Wide Web, 2014, pp. 1157–1162.

[29] Q. Lin, Y. Ma, J. Chen, Q. Zhu, C.A. Coello Coello, K.-C. Wong, F. Chen, An adaptive immune-inspired multi-objective algorithm with multiple differential evolution strategies, Inf. Sci. 430–431 (2018) 46–64.

[30] W. Zheng, Z. Liao, Using a heuristic approach to design personalized tour routes for heterogeneous tourist groups, Tour. Manag. 72 (2019) 313–325.

[31] C.A. Coello Coello, Evolutionary Algorithms for Solving Multi-Objective Problems, Springer, New York, NY, 2007.

[32] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.

[33] I. Csiszár, I-divergence geometry of probability distributions and minimisation problems, Ann. Probab. (1975) 146–158.

[34] S. Mirjalili, S. Saremi, M.S. Mirjalili, S.D.L. Coelho, Multi-objective grey wolf optimiser: A novel algorithm for multi-criterion optimisation, Expert Syst. Appl. 47 (2016) 106–119.

[35] C.A. Coello Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimisation, IEEE Trans. Evol. Comput. 8 (3) (2004) 256–279.

[36] J.D. Knowles, D.W. Corne, Approximating the nondominated front using the Pareto archived evolution strategy, Evol. Comput. 8 (2) (2000) 149–172.

[37] J. Bader, E. Zitzler, Hype: An algorithm for fast hypervolume-based many-objective optimisation, Evol. Comput. 19 (1) (2011) 45–76.

[38] V.A. Traag, L. Waltman, N.J. van Eck, From Louvain to Leiden: Guaranteeing well-connected communities, Sci. Rep. 9 (2019) 5233.

[39] D. Lusseau, K. Schneider, O.J. Boisseau, P. Haase, E. Slooten, S.M. Dawson, The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations, Behav. Ecol. Sociobiol. 54 (4) (2003) 396–405.

[40] A.R. Rossi, K.N. Ahmed, The network data repository with interactive graph analytics and visualization, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2015.

[41] J. Leskovec, J. Kleinberg, C. Faloutsos, Graph evolution: Densification and shrinking diameters, ACM Trans. Knowl. Discov. Data 1 (1) (2007) 1–40.

[42] H. Yin, R.A. Benson, J. Leskovec, F.D. Gleich, Local higher-order graph clustering, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 555–564.

[43] J. Leskovec, J. McAuley, Learning to discover social circles in ego networks, in: Advances in Neural Information Processing Systems, vol. 25, 2012.

[44] J. Leskovec, D. Huttenlocher, J. Kleinberg, Signed networks in social media, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2010, pp. 1361–1370.
[45] J. Leskovec, D. Huttenlocher, J. Kleinberg, Predicting positive and negative links in online social networks, in: Proceedings of the 19th International Conference on World Wide Web, 2010, pp. 641–650.
[46] B. Rozemberczki, R. Sarkar, Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, CIKM'20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 1325–1334.
[47] X. Zhou, X. Zhao, Y. Liu, A multiobjective discrete bat algorithm for community detection in dynamic networks, Appl. Intell. 48 (2018) 3081–3093.
[48] C.A. Coello Coello, S.M. Lechuga, MOPSO: A proposal for multiple objective particle swarm optimisation, in: Proceedings of the IEEE Congress on Evolutionary Computation, CEC, vol. 2, 2002, pp. 1051–1056.
[49] M. Gong, J. Yan, B. Shen, L. Ma, Q. Cai, Influence maximisation in social networks based on discrete particle swarm optimisation, Inf. Sci. 367–368 (2016) 600–614.
[50] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[51] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength pareto evolutionary algorithm, 2001.
[52] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable multi-objective optimisation test problems, in: Proceedings of the IEEE Congress on Evolutionary Computation, CEC, vol. 1, 2002, pp. 825–830.
[53] J.R. Schott, Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimisation, Tech. Rep., 1995.
[54] K. Deb, Multi-objective optimisation using evolutionary algorithms: An introduction, in: L. Wang, A. Ng, K. Deb (Eds.), Multi-Objective Evolutionary Optimisation for Product Design and Manufacturing, Springer, London, 2011, pp. 3–34.
[55] X.-G. Zhou, G.-J. Zhang, Abstract convex underestimation assisted multistage differential evolution, IEEE Trans. Cybern. 47 (9) (2017) 2730–2741.