Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.090149

ARTICLE

Distributed Potential-Game Optimization with Local Exchanges for Dynamic Maximum-Weight Independent Sets

Changlei Zhan[1] **and Yong Ding**[2,*]

[1]School of Information Science and Technology, Fudan University, Shanghai, 200240, China
[2]School of Information Engineering, Nanchang University, Nanchang, 330031, China

*Corresponding Author: Yong Ding. Email: dingyong1014@hotmail.com

Received: 31 July 2026; Accepted: 16 September 2026; Published: 22 October 2026

ABSTRACT: Maximum-weight independent-set (MWIS) selection is a core primitive in distributed intelligent systems because interference-limited communication, collision-free sensing, task activation, and resource allocation can all be represented by weighted conflict graphs. Existing distributed maximal-independent-set routines scale well but can lock in low-weight choices, whereas centralized MWIS solvers require global information and become impractical on large or time-varying networks. This paper develops an exchange-refined adaptive distributed game algorithm (ADGA-X) for static and dynamic weighted conflict networks. Unlike a plain distributed greedy rule or an unconstrained potential-game iteration, ADGA-X uses a penalty threshold that makes conflict removal locally dominant, then couples certified parallel better responses with bounded-radius exchange refinement and dynamic repair. First, a penalty-dominant binary game is constructed whose local utilities admit an exact potential. A common upper bound on vertex weights gives a penalty choice implementable with distributed information, makes every pure Nash equilibrium a feasible maximal independent set, and makes every global potential maximizer an MWIS solution. Second, nonadjacent strict better responses are scheduled in parallel by lexicographic one-hop priorities, preserving additive potential improvement and finite convergence. Third, one- and two-vertex exchanges use bounded-radius proposal coordination to refine the equilibrium without a global controller. Exhaustive refinement terminates at full one- and two-vertex exchange stability; the practical candidate- and round-capped variant remains feasible, monotone, finitely convergent, and singleton-exchange stable, thereby retaining a degree-dependent approximation guarantee. Fourth, an explicit warm-start repair is introduced for topology and weight changes, together with bounds on newly created conflicts and potential drift. Paired 20-instance validation shows that ADGA-X significantly outperforms the tested belief-propagation (BP) decoder on Erdős–Rényi (ER) and random geometric graph (RGG) instances but not on Barabási–Albert-type (BA) graphs, clarifying the topology-dependent tradeoff. Across these three graph families, ADGA-X improves the base potential-game equilibrium by 8.19%–11.61%, and every returned set is feasible by construction. On exact small instances, it reaches a mean optimality ratio of 0.990, and the repaired warm start reduces post-change recovery outer rounds by approximately 47.0% while maintaining a mean warm/cold final-weight ratio of 1.0020.

KEYWORDS: Potential game; maximum-weight independent set (MWIS); dynamic distributed optimization; local exchange; exact penalty; dynamic graph

1 Introduction

Distributed intelligent systems increasingly depend on local agents that must coordinate under limited communication, incomplete topology information, and changing operating conditions. Representative examples include wireless link scheduling, vehicular content retrieval, sensor sleep scheduling,

Copyright © 2026 The Authors. Published by Tech Science Press. This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

edge-computing service admission, and cooperative task allocation. In these applications, mutually interfering activities cannot be activated simultaneously, while heterogeneous priorities, energy levels, deadlines, or service values must be respected. A weighted conflict graph gives a common abstraction: vertices represent candidate activities, edges encode pairwise incompatibility, and weights quantify benefits. Selecting a compatible set with maximum total benefit is the maximum-weight independent-set (MWIS) problem [1,2]. Vehicular retrieval, coordinated wireless access, and wireless-powered edge computing are representative settings in which local interference and heterogeneous rewards lead naturally to weighted conflict decisions [3–5]. Concretely, a vertex may denote a candidate link, sensing task, vehicle request, or admitted edge job; an edge denotes a simultaneous-use conflict; and the weight can encode throughput, urgency, utility, or an application-specific priority.

MWIS is NP-hard on general graphs, and the gap between exact solution quality and distributed implementability is substantial. Recent learning-based graph-optimization research and Boolean-network formulations illustrate continuing interest in scalable independent-set search [6,7]. Exact, reduction-based, and hardware-assisted approaches can exploit sophisticated global search [8–10], but repeatedly collecting a complete graph is costly and fragile in large or time-varying systems. This motivates algorithms whose routine updates use one-hop state exchange and whose optional refinements require only bounded-radius coordination.

Classical distributed maximal-independent-set algorithms provide scalable symmetry breaking [11], but maximality does not imply high total weight. A low-weight vertex selected early can block several valuable neighbors, and a one-pass suppression rule cannot repair that choice. Message-passing methods exploit richer soft information and can be effective on favorable graph families, yet loopy graphs may require additional decoding or repair [2]. Greedy methods remain attractive, although their performance is sensitive to ordering and local score design [12]. Classical local-search schemes further improve independent sets through one-to-two and larger swap neighborhoods [13–15].

Potential games provide a complementary route because unilateral payoff improvements can be certified by a global scalar function [16]. Closely related graph-combinatorial work has used potential-game dynamics for distributed minimum-vertex-cover optimization, while networked applications include aerial target assignment and risk-aware congestion control [17–19]. The latter study appeared in Computers, Materials & Continua (CMC); the journal name is written in full here at the first use of the abbreviation. These methods provide useful local decision mechanisms, but they do not by themselves supply the MWIS-specific combination of an exact feasibility threshold, conflict-free parallel updates, swap-based refinement, and snapshot repair. The MWIS setting therefore raises two specific difficulties: a poorly calibrated utility may admit conflicting equilibria, and a feasible unilateral equilibrium may have low weight because no single vertex can replace a blocking cluster.

This paper addresses both difficulties through an exchange-refined game-theoretic distributed optimization method, denoted ADGA-X. The framework combines an exact-potential formulation, conflict-free parallel updates, local exchanges, and warm-start repair after topology or weight changes. The contribution is primarily the rigorously analyzed integration of these components for distributed MWIS rather than a claim that the classical potential-game concept or the one-/two-swap neighborhood is new in isolation. It is optimization-based rather than training-dependent and can therefore operate without historical data or act as a deterministic feasibility layer beneath a learning-assisted system.

The main contributions are summarized as follows.

1. A penalty-dominant exact potential game is developed for MWIS. We prove that a penalty exceeding a common upper bound on the vertex weights excludes every conflicting equilibrium, makes each

equilibrium a feasible maximal independent set, and makes every global potential maximizer an MWIS solution. The bound can be specified from system limits or obtained by a preliminary max-consensus.
2. A one-hop parallel better-response mechanism is introduced. A distributed priority rule selects an independent updater set, so simultaneous local gains add exactly to the potential gain. Finite convergence and an explicit state-transition bound are established; the exchange refinement uses bounded-radius messages relayed over one-hop links.
3. A distributed exchange stage allows one or two unselected vertices to replace their selected conflict set. Compatible positive-gain proposals can be executed in parallel. Exhaustive refinement reaches full one- and two-vertex exchange stability; the practical candidate- and round-capped implementation finishes with exhaustive singleton polishing. Both variants terminate finitely, and singleton-exchange stability implies a $1/\Delta$ worst-case weight guarantee for $\Delta \geq 1$; the edgeless case is exact.
4. A warm-start mechanism is formulated for dynamic graphs. Bounds are derived for the number of newly violated edges and the potential drift caused by topology, weight, and penalty changes.
5. Reproducible experiments compare ADGA-X with synchronous distributed greedy, ten-restart randomized greedy, belief propagation, and the unrefined potential-game dynamics. The results quantify solution weight, feasibility, exact-optimum ratio, density sensitivity, scaling, and post-change recovery.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 defines the static and dynamic MWIS models. Section 4 develops the exact potential game. Section 5 presents ADGA-X and its theoretical guarantees. Section 6 analyzes dynamic repair and implementation complexity. Section 7 reports the experiments, and Section 8 concludes the paper.

2 Related Work

2.1 Distributed Independent-Set Optimization

An independent set is locally checkable and is therefore a canonical object in distributed computing. Luby's parallel randomized method constructs a maximal independent set using local communication [11]. Such methods primarily target maximality or cardinality and do not directly optimize heterogeneous vertex weights. Weighted variants are commonly approached through greedy rankings, local ratios, message passing, or centralized reductions. Message passing for MWIS derives directed-edge messages and has useful interpretations on trees and sparse graphs [2]; on general loopy networks, a feasible decoding or repair step can still be necessary. Recent work has revisited independent-set search through graph augmentation, Boolean-network generalizations, and specialized optimization hardware [6–8]. Exact branch-and-reduce methods can solve large weighted instances with global reductions, but they assume centralized graph access [10].

Classical local-search methods show that swap neighborhoods can substantially improve a maximal independent set. Andrade, Resende, and Werneck developed fast local search around one-to-two and larger exchanges, and later large-scale solvers combined local improvement with evolutionary or kernelization mechanisms [13–15]. ADGA-X does not claim the swap neighborhood itself as new; its contribution is to realize selected one-/two-vertex exchanges through bounded-radius proposal construction and a conflict-free distributed compatibility schedule, while preserving feasibility and monotone improvement.

2.2 Potential Games, Learning, and Distributed Optimization

A finite game is an exact potential game when every unilateral utility difference equals the change of a common potential [16]. This property yields pure equilibria and finite termination of strict better-response paths. Of particular relevance, a self-stabilizing network algorithm formulates minimum vertex cover

through a potential game [17]; other networked applications use potential-game ideas for heterogeneous aerial target assignment and risk-aware distributed congestion control [18,19].

Learning-assisted MWIS schedulers provide a different quality–adaptation tradeoff. For example, distributed graph neural network (GNN) scheduling learns topology-aware embeddings before a greedy MWIS stage and can improve a fixed heuristic when representative training data are available [20]. Such approaches may generalize well, but their guarantees depend on the learned model and deployment distribution; ADGA-X instead requires no offline training and keeps an explicit feasibility/monotonicity certificate. Likewise, the alternating direction method of multipliers (ADMM) is a powerful distributed framework for continuous convex programs [21]. Direct use for binary MWIS normally requires relaxation, splitting, or nonconvex penalties, whereas ADGA-X operates on the native $\{0,1\}$ action space and never returns a conflicting set after the feasible stage. This discrete guarantee is obtained at the cost of combinatorial local-search overhead and a graph-dependent approximation bound.

2.3 Research Gap and Positioning

The base penalty game for MWIS is conceptually simple, but three issues prevent it from being a complete distributed optimization method. A penalty threshold must be linked explicitly to feasibility; parallel updates must preserve the unilateral potential logic; and local equilibria must be refined without destroying distributed execution. Existing maximal-independent-set methods do not provide the desired weight-oriented exchange certificate, whereas generic potential-game learning does not exploit the combinatorial structure of an independent set. ADGA-X fills this gap by combining exact-penalty design, conflict-free parallel updates, local exchanges, and dynamic warm starting in one framework. Table 1 summarizes the main distinctions among representative MWIS solution paradigms and highlights the information, feasibility, quality, and dynamic-reuse properties of ADGA-X.

Table 1: Positioning of representative maximum-weight independent-set (MWIS) solution paradigms.

Paradigm	Typical Information	Feasibility Mechanism	Quality Mechanism	Dynamic Reuse
Distributed greedy/MIS	One-hop state/score	Constructive suppression	Ordering/local score	Usually rerun
Message passing	Directed edge messages	Decode/repair on loops	Soft local messages	Re-iterate messages
Learned GNN scheduler	Features plus trained model	Greedy/repair layer	Learned topology embedding	Inference, retraining if shifted
Centralized reduction/local search	Global graph	Explicit centralized set operations	Reductions and swaps	Global update/restart
ADGA-X	One-hop state plus bounded-radius proposals	Exact penalty and feasible exchanges	Potential ascent plus local swaps	Local repair and warm start

Note: MWIS: maximum-weight independent set; MIS: maximal independent set; GNN: graph neural network; ADGA-X: exchange-refined adaptive distributed game algorithm.

3 Problem Formulation

3.1 Weighted Conflict Graph

Let an undirected graph be

$$G = (V, E), \qquad V = \{1, 2, \ldots, n\}, \qquad E \subseteq \{\{u, v\} : u \neq v\}. \tag{1}$$

For vertex v, its open neighborhood and degree are

$$N(v) = \{u \in V : \{u, v\} \in E\}, \qquad d_v = |N(v)|, \tag{2}$$

and the maximum degree is

$$\Delta = \max_{v \in V} d_v. \tag{3}$$

Each vertex has a strictly positive weight

$$0 < w_v \leq \bar{w}, \qquad w_{\max} = \max_{v \in V} w_v \leq \bar{w}, \tag{4}$$

where zero-weight vertices can be removed without changing the optimum. The common bound $\bar{w}$ may follow from an application-level weight range. When it is unavailable and the conflict graph is connected, the nodes can obtain $w_{\max}$ by finite max-consensus before optimization. For a disconnected graph, max-consensus may be run independently in each connected component and the components may use their own valid penalties because both the game and the MWIS objective decompose. For notational simplicity, the analysis below uses one common bound $\bar{w}$.

A set $S \subseteq V$ is independent if

$$\{u, v\} \notin E, \qquad \forall u, v \in S, \tag{5}$$

and its total weight is

$$W(S) = \sum_{v \in S} w_v. \tag{6}$$

The set form of MWIS is

$$S^\star \in \arg\max_{S \subseteq V} \{W(S) : S \text{ satisfies (5)}\}. \tag{7}$$

Let $x_v \in \{0, 1\}$ indicate selection. The binary program is

$$\max_{x \in \{0,1\}^n} \quad \sum_{v \in V} w_v x_v, \tag{8}$$

$$\text{s.t.} \quad x_u + x_v \leq 1, \qquad \forall \{u, v\} \in E. \tag{9}$$

In applications, $x_v = 1$ may mean that a sensor transmits, a radio-frequency identification (RFID) reader is activated, an edge task is admitted, or a robot is assigned a mutually exclusive operation. Wireless scheduling, vehicular retrieval, and edge admission provide typical sources of such conflict graphs [3–5]. The edge set can be formed from interference thresholds, resource-overlap rules, or mutually exclusive task constraints, while w_v can be measured or predicted from local throughput, deadline, reliability, energy, or

service-value information. Thus, the objective in (8) aggregates local benefit, the variables are local activation decisions, and (9) is the physical incompatibility condition enforced by the game and exchange stages below.

3.2 Exact-Penalty Representation

For a binary vector $\boldsymbol{x}$, define the number of selected conflict edges as

$$C_E(\boldsymbol{x}) = \sum_{\{u,v\}\in E} x_u x_v. \tag{10}$$

Given a penalty $\lambda > 0$, consider

$$F_\lambda(\boldsymbol{x}) = \sum_{v\in V} w_v x_v - \lambda C_E(\boldsymbol{x}). \tag{11}$$

If $\boldsymbol{x}$ is feasible, then $C_E(\boldsymbol{x}) = 0$ and $F_\lambda(\boldsymbol{x})$ equals its selected weight. The penalty will later be chosen to make every infeasible state locally removable.

3.3 Dynamic Graph Model

A dynamic distributed system is represented by epochs $k = 0, 1, 2, \ldots$ with

$$G^k = (V^k, E^k), \qquad \boldsymbol{w}^k = (w_v^k)_{v\in V^k}. \tag{12}$$

For clarity, the analysis first assumes a fixed vertex set $V^k = V$; node arrivals and departures can be handled by adding new vertices with action zero and deleting departed states. The topology variation is measured by

$$\delta_E^k = |E^k \triangle E^{k-1}|, \tag{13}$$

where $\triangle$ is the symmetric difference. The weight variation is measured by

$$\delta_w^k = \|\boldsymbol{w}^k - \boldsymbol{w}^{k-1}\|_1. \tag{14}$$

The objective at epoch k is

$$S_k^\star \in \operatorname*{arg\,max}_{S \text{ independent in } G^k} \sum_{v\in S} w_v^k. \tag{15}$$

Instead of discarding the previous solution, the proposed method repairs and refines it using local information. For the epoch-wise guarantees, the graph snapshot is assumed fixed while one repair/optimization episode is executed. The changes need not be infinitesimal, but warm starting is most useful when δ_E^k and δ_w^k are localized enough that many previous decisions remain unaffected. If the environment changes faster than an episode can settle, feasibility can still be repaired on the latest observed snapshot, but the equilibrium and exchange-stability statements should then be interpreted snapshot by snapshot rather than as a tracking theorem for continuously changing graphs.

4 Penalty-Dominant Exact Potential Game

4.1 Strategic Model

Associate one player with each vertex. The finite strategic game is

$$\mathcal{G} = \left(V, \{\mathcal{A}_v\}_{v\in V}, \{U_v\}_{v\in V}\right), \qquad \mathcal{A}_v = \{0,1\}. \tag{16}$$

The action profile is $\boldsymbol{a} = (a_v)_{v\in V}$, and the selected set is

$$S(\boldsymbol{a}) = \{v \in V : a_v = 1\}. \tag{17}$$

For a penalty satisfying

$$\lambda > \bar{w} \geq w_{\max}, \tag{18}$$

the local utility is defined as

$$U_v(\boldsymbol{a}) = w_v a_v - \lambda a_v \sum_{u\in N(v)} a_u. \tag{19}$$

The first term rewards valuable activation. The second makes a selected vertex pay once for every selected neighbor. Although each conflict is represented in the utilities of both endpoints, the potential below counts each undirected edge once. In practice, $\bar{w}$ may be a certified system ceiling (for example, the largest admissible priority code) or the result of max-consensus. A modest margin such as $\lambda = 1.05\,\bar{w}$ protects against small numerical/quantization error while avoiding an unnecessarily large coefficient. Overestimating $\bar{w}$ is safe; underestimating it is safe only as long as the implemented λ still exceeds the true $w_{\max}$. For a connected synchronous graph, standard max-consensus reaches the network maximum in at most D_G rounds, where D_G is the graph diameter; this one-time calibration is reported separately from the optimization-round counts.

4.2 Exact Potential Property

Define

$$\Phi(\boldsymbol{a}) = \sum_{v\in V} w_v a_v - \lambda \sum_{\{u,v\}\in E} a_u a_v. \tag{20}$$

Theorem 1 (Exact potential): *The game in (16)–(19) is an exact potential game with potential (20).*

Proof: Fix v and all actions $\boldsymbol{a}_{-v}$. Let

$$c_v(\boldsymbol{a}_{-v}) = \sum_{u\in N(v)} a_u. \tag{21}$$

For two actions $a_v, a_v' \in \{0,1\}$, the utility difference is

$$U_v(a_v, \boldsymbol{a}_{-v}) - U_v(a_v', \boldsymbol{a}_{-v}) = (a_v - a_v')\big(w_v - \lambda c_v(\boldsymbol{a}_{-v})\big). \tag{22}$$

Only the vertex term of v and edge terms incident to v change in the potential, giving

$$\Phi(a_v, \boldsymbol{a}_{-v}) - \Phi(a_v', \boldsymbol{a}_{-v}) = (a_v - a_v')\big(w_v - \lambda c_v(\boldsymbol{a}_{-v})\big). \tag{23}$$

Eqs. (22) and (23) are identical. □

The local gain from activation is

$$\Delta_v(\boldsymbol{a}) = U_v(1, \boldsymbol{a}_{-v}) - U_v(0, \boldsymbol{a}_{-v}) = w_v - \lambda c_v(\boldsymbol{a}_{-v}). \tag{24}$$

A tie-preserving best response is therefore

$$b_v(\boldsymbol{a}) = \begin{cases} 1, & \Delta_v(\boldsymbol{a}) > 0, \\ 0, & \Delta_v(\boldsymbol{a}) < 0, \\ a_v, & \Delta_v(\boldsymbol{a}) = 0. \end{cases} \tag{25}$$

Under the standing assumption $w_v > 0$ and $\lambda > w_{\max}$, the equality branch in (25) is not reached: if $c_v = 0$, then $\Delta_v = w_v > 0$, whereas if $c_v \geq 1$, then $\Delta_v < 0$. We retain the tie-preserving branch only to make the rule robust to extensions that admit zero weights or finite-precision ties. Moreover, for every admissible $\lambda > w_{\max}$ the sign pattern of Δ_v and hence the local best-response map and equilibrium set are identical. For the empty or repaired feasible starts used by ADGA-X, every eligible activation has $c_v = 0$ and gain w_v, so the implemented Stage-I schedule is also independent of the precise admissible λ. From an arbitrary conflicting initial profile, however, the gain magnitude $|\Delta_v|$ in (31) can reorder deactivating vertices and thus change the transient schedule, although feasibility and convergence guarantees are unaffected. Stage II is evaluated on feasible sets and is independent of λ altogether.

4.3 Feasibility and Optimality Properties

Theorem 2 (Equilibrium feasibility and maximality): *Under (18), every pure Nash equilibrium induces a feasible maximal independent set.*

Proof: Suppose a pure Nash equilibrium $\boldsymbol{a}$ contains a selected edge $\{u, v\}$. Then $c_v \geq 1$. The utility gain when v leaves is

$$U_v(0, \boldsymbol{a}_{-v}) - U_v(1, \boldsymbol{a}_{-v}) = \lambda c_v - w_v \geq \lambda - w_v > 0, \tag{26}$$

contradicting equilibrium. Hence $S(\boldsymbol{a})$ is independent. If an unselected vertex v had no selected neighbor, then

$$U_v(1, \boldsymbol{a}_{-v}) - U_v(0, \boldsymbol{a}_{-v}) = w_v > 0, \tag{27}$$

which is another profitable deviation. Thus every unselected vertex is adjacent to a selected vertex, so the independent set is maximal. □

Proposition 1 (Exact-penalty global optimum): *Every global maximizer of (20) under (18) is an MWIS solution.*

Proof: If a profile is infeasible, choose a selected vertex v with $c_v \geq 1$. Removing it changes the potential by

$$\Phi(0, \boldsymbol{a}_{-v}) - \Phi(1, \boldsymbol{a}_{-v}) = \lambda c_v - w_v > 0. \tag{28}$$

Therefore no infeasible profile can maximize Φ. On every feasible profile, the penalty term is zero and

$$\Phi(\boldsymbol{a}) = W(S(\boldsymbol{a})). \tag{29}$$

Maximizing the potential is consequently equivalent to solving (7). □

Corollary 1. *The game has at least one pure Nash equilibrium.*

Proof: The finite action set contains a maximizer of Φ. By Theorem 1, a potential maximizer cannot admit a profitable unilateral deviation and is therefore a pure Nash equilibrium. □

The distinction between Proposition 1 and Theorem 2 is important. Global potential maximization is exact but generally hard; local better responses are distributed and convergent but can stop at a suboptimal maximal independent set. The exchange stage introduced next narrows this gap. The core static guarantees require positive bounded weights, a penalty strictly above the true maximum weight, reliable exchange of the stated local messages, and a graph that is unchanged during one optimization episode. The convergence proofs do not require the random component of the priority to be continuous because the deterministic identifier already makes priorities strict.

5 Exchange-Refined Adaptive Distributed Game Algorithm

5.1 Parallel Better-Response Stage

At round t, node v evaluates $b_v(t) = b_v(\boldsymbol{a}(t))$. The set of vertices with a strict state change is

$$\mathcal{E}(t) = \{v \in V : b_v(t) \neq a_v(t)\}. \tag{30}$$

Each eligible node assigns itself the priority

$$\pi_v(t) = \big(|\Delta_v(\boldsymbol{a}(t))|, \rho_v(t), -v\big), \tag{31}$$

where $\rho_v(t)$ is drawn independently from a continuous distribution on $(0,1)$, and tuples are compared lexicographically. The final index component already guarantees a unique priority for adjacent eligible vertices, so randomization is not needed for Lemma 1 or Theorem 3; $\rho_v(t)$ is retained to reduce persistent bias toward the same identifiers when equal gains recur. The updater set is

$$\mathcal{I}(t) = \{v \in \mathcal{E}(t) : \pi_v(t) > \pi_u(t),\ \forall u \in N(v) \cap \mathcal{E}(t)\}. \tag{32}$$

The lexicographic rule gives every adjacent eligible pair a unique priority, and $\mathcal{I}(t)$ is independent. The state update is

$$a_v(t+1) = \begin{cases} b_v(t), & v \in \mathcal{I}(t), \\ a_v(t), & v \notin \mathcal{I}(t). \end{cases} \tag{33}$$

For $v \in \mathcal{I}(t)$, define its accepted gain as

$$g_v(t) = U_v(a_v(t+1), \boldsymbol{a}_{-v}(t)) - U_v(a_v(t), \boldsymbol{a}_{-v}(t)) > 0. \tag{34}$$

Lemma 1 (Additive parallel improvement): *For the independent updater set (32),*

$$\Phi(\boldsymbol{a}(t+1)) - \Phi(\boldsymbol{a}(t)) = \sum_{v \in \mathcal{I}(t)} g_v(t) > 0 \tag{35}$$

whenever $\mathcal{I}(t) \neq \varnothing$.

Proof: No two updating vertices are adjacent. Hence an update by one member of $\mathcal{I}(t)$ does not alter the neighbor-action sum used by another member. The joint move can be decomposed into unilateral moves in any order, each retaining gain $g_v(t)$. Exact potential differences then sum to (35). □

Theorem 3 (Finite convergence of Stage I): *Starting from any action profile, the parallel better-response stage reaches a pure Nash equilibrium in finitely many active rounds for every realization of the priority tie breaker.*

Proof: At every non-equilibrium profile, $\mathcal{E}(t)$ is nonempty. The largest-priority eligible vertex belongs to $\mathcal{I}(t)$, so at least one strict change occurs. Lemma 1 gives a strict potential increase. Since there are only 2^n action profiles, a strictly increasing sequence cannot revisit a state. Thus the number of active state transitions satisfies

$$T_{\mathrm{BR}} \le 2^n - 1. \tag{36}$$

The terminal state has no strict better response and is a pure Nash equilibrium. □

The bound (36) is intentionally worst-case and does not claim polynomial-time solution of an NP-hard problem. A more informative pseudo-polynomial bound is available when weights and the implemented penalty are integer-valued. Let $g_{\min} > 0$ be the smallest realizable strict potential gain. Since $-\lambda|E| \le \Phi \le n\bar{w}$, any strict-improvement trajectory satisfies

$$T_{\mathrm{BR}} \le \frac{n\bar{w} + \lambda|E|}{g_{\min}}. \tag{37}$$

For the feasible empty or repaired starts used by ADGA-X with integer weights, $\Phi = W(S) \in [0, n\bar{w}]$ and $g_{\min} \ge 1$, so $T_{\mathrm{BR}} \le n\bar{w}$. This remains a worst-case bound, but it is substantially more informative than $2^n - 1$ for bounded integer weights. In sparse experiments, many nonadjacent updates occur simultaneously, and observed round counts are much smaller.

5.2 One- and Two-Vertex Exchange Stage

Let S be the feasible independent set returned by Stage I. A proposal is a set of currently unselected vertices

$$P \subseteq V \backslash S, \qquad 1 \le |P| \le 2, \tag{38}$$

that is internally independent:

$$\{p, q\} \notin E, \qquad \forall p, q \in P. \tag{39}$$

Its selected conflict set is

$$C_S(P) = S \cap \bigcup_{p \in P} N(p). \tag{40}$$

Replacing all conflicting selected vertices by P gives

$$S \oplus P = \big(S \backslash C_S(P)\big) \cup P. \tag{41}$$

Because every selected neighbor of a proposed vertex is removed and P is independent, $S \oplus P$ is feasible. The exchange gain is

$$\Gamma_S(P) = \sum_{p \in P} w_p - \sum_{u \in C_S(P)} w_u. \tag{42}$$

A proposal is improving when

$$\Gamma_S(P) > 0. \tag{43}$$

To execute multiple proposals in one round, let the affected set be

$$A_S(P) = P \cup C_S(P). \tag{44}$$

Proposals P_i and P_j are compatible if

$$A_S(P_i) \cap A_S(P_j) = \varnothing, \tag{45}$$

$$\{p, q\} \notin E, \quad \forall p \in P_i,\ q \in P_j. \tag{46}$$

Each positive proposal receives a strict lexicographic priority based on its gain, inserted weight, an independent continuous tie breaker, and a deterministic proposal identifier. Two proposals are adjacent in a proposal-conflict graph when they violate either (45) or (46). In each contention micro-round, every active proposal whose priority exceeds those of all active conflicting proposals is accepted; accepted proposals and their conflicting neighbors then retire. Repeating this local rule produces a maximal compatible collection $\mathcal{P}$. The descending-priority scan used by the synchronous simulator is an exact emulation of these local micro-rounds, not a centralized information requirement. To see this, order proposals by the same strict priority. The highest active proposal has no higher-priority active conflict and is therefore accepted in the current micro-round; deleting it and all conflicting proposals gives exactly the residual active set produced by the corresponding greedy scan. Induction on the residual proposal-conflict graph proves that both procedures return the same maximal compatible collection. Its parallel update is

$$S^+ = \left(S \setminus \bigcup_{P \in \mathcal{P}} C_S(P)\right) \cup \left(\bigcup_{P \in \mathcal{P}} P\right). \tag{47}$$

Lemma 2 (Parallel exchange gain): *If all proposals in $\mathcal{P}$satisfy (43), (45), and (46), then S^+ is independent and*

$$W(S^+) - W(S) = \sum_{P \in \mathcal{P}} \Gamma_S(P) > 0. \tag{48}$$

Proof: Internal and cross-proposal independence exclude edges among newly added vertices. Every edge from a new vertex to an old selected vertex ends at its own conflict set, which is removed. Disjoint affected sets prevent double counting, so the weight difference separates into the sum of proposal gains. □

Theorem 4 (Finite convergence of exhaustive Stage II): *Suppose all admissible singleton and two-vertex proposals are available and the scheduler accepts at least one compatible positive-gain proposal whenever one exists. Then repeated exchanges terminate at a one- and two-vertex exchange-stable independent set after finitely many exchange rounds.*

Proof: Every accepted round strictly increases $W(S)$ by Lemma 2. There are finitely many independent sets. Therefore no set can repeat and the procedure terminates. Under exhaustive proposal generation and the stated progress condition, termination can occur only when no admissible singleton or pair proposal has positive gain. □

Definition 1 (Singleton-exchange stability): *An independent set S is singleton-exchange stable when*

$$w_v \le \sum_{u \in S \cap N(v)} w_u, \qquad \forall v \in V \backslash S. \tag{49}$$

Stage II termination implies (49); it additionally enforces the stronger pair inequalities

$$w_p + w_q \le \sum_{u \in C_S(\{p,q\})} w_u \tag{50}$$

for every nonadjacent pair $p, q \notin S$ considered by the full exchange rule.

Theorem 5 *(Degree-dependent guarantee): Let S be singleton-exchange stable and let $S^\star$ be an MWIS. If $\Delta \ge 1$, then*

$$W(S) \ge \frac{1}{\Delta} W(S^\star). \tag{51}$$

For $\Delta = 0$, both S and $S^\star$ equal V.

Proof: Write $A = S \cap S^\star$, $B = S \backslash S^\star$, and $C = S^\star \backslash S$. Because $S^\star$ is independent, no vertex in A is adjacent to a vertex in C. Thus singleton stability gives, for every $v \in C$,

$$w_v \le \sum_{u \in B \cap N(v)} w_u. \tag{52}$$

Summing over C and observing that every $u \in B$ is adjacent to at most Δ vertices of C yields

$$W(C) \le \Delta W(B). \tag{53}$$

Consequently, for $\Delta \ge 1$,

$$\begin{aligned} W(S^\star) &= W(A) + W(C) \\ &\le W(A) + \Delta W(B) \le \Delta\big(W(A) + W(B)\big) = \Delta W(S), \end{aligned} \tag{54}$$

which proves (51). If $\Delta = 0$, the graph has no edges; positive weights and singleton stability force $S = V$, which is optimal. □

The theorem is conservative, especially on weighted random graphs. Pair exchanges improve empirical quality beyond singleton stability, although a sharper general bound would require additional graph structure. The one-/two-vertex neighborhood is a classical local-search idea [13–15]; the novelty claimed here is the bounded-radius distributed proposal construction, compatibility scheduling, and its integration with the exact-potential and dynamic-repair layers.

Once no improving singleton remains, every genuinely new improving pair must share at least one selected blocker. Indeed, if $C_S(\{p\}) \cap C_S(\{q\}) = \varnothing$, then

$$\Gamma_S(\{p, q\}) = \Gamma_S(\{p\}) + \Gamma_S(\{q\}), \tag{55}$$

so a positive pair gain would imply a positive singleton gain. Thus, after singleton stability has been reached, restricting pair generation to candidates sharing a selected blocker is lossless for positive pairs. Before that point, the practical mixed stage uses the same restriction as a scalable heuristic while retaining every positive singleton proposal; the unrestricted pair-stability claim is therefore reserved for the exhaustive theoretical variant.

5.3 ADGA-X Procedure

Table 2 summarizes the full procedure. The theoretical pair stage can enumerate all locally constructible pairs. For predictable runtime, the implementation retains at most q high-weight unselected neighbors per selected blocker and permits at most R mixed singleton/pair outer rounds; the experiments use $q = 15$ and $R = 25$. It then performs singleton-only polishing until no profitable singleton proposal remains. Every accepted move preserves feasibility and strictly increases weight, while the final polishing guarantees singleton-exchange stability, so Theorem 5 still applies. If the pair budget is exhausted, no unrestricted or q-restricted pair-stability claim is made. Thus, exhaustive ADGA-X has feasibility, monotonicity, finite convergence, singleton stability, and full one-/two-vertex exchange stability; capped ADGA-X keeps feasibility, monotonicity, finite convergence, singleton stability, and Theorem 5, but not full pair stability.

Table 2: Exchange-refined adaptive distributed game algorithm (ADGA-X).

Step	Distributed Operation
1	Initialize $a_v \leftarrow 0$ (or a repaired warm start), select $\lambda > \bar{w} \geq w_{\max}$, and exchange one-hop actions.
2	Each node computes c_v, Δ_v, b_v, and priority π_v using (21), (24), (25), and (31).
3	Eligible local priority winners form $\mathcal{I}(t)$ and update by (33). Repeat until no strict better response exists.
4	Each unselected node constructs its singleton proposal. Candidate pairs are formed from nonadjacent unselected vertices sharing a selected blocker, using radius-two information relayed through that blocker.
5	Nodes compute $C_S(P)$ and $\Gamma_S(P)$. Strict proposal priorities drive repeated local contention micro-rounds on the proposal-conflict graph.
6	The resulting maximal compatible collection is executed by (47). Repeat outer exchange rounds until no evaluated proposal remains or the cap R is reached.
7	Exhaust singleton proposals and output a feasible singleton-exchange-stable set S; restricted-pair stability is additionally obtained only when Step 6 terminates naturally.

6 Dynamic Repair and Distributed Complexity

6.1 Warm-Start Repair

Suppose S^{k-1} is independent in G^{k-1}. New edges can create conflicts in the old set. Define the violated new-edge set

$$\mathcal{V}^k = \left\{ \{u, v\} \in E^k \backslash E^{k-1} : u, v \in S^{k-1} \right\}. \tag{56}$$

It immediately satisfies

$$|\mathcal{V}^k| \leq |E^k \backslash E^{k-1}| \leq \delta_E^k. \tag{57}$$

Thus only endpoints of newly conflicting edges require immediate repair. Assign each old selected vertex the lexicographic priority

$$\eta_v^k = \left(\sigma_v^k, v\right), \qquad \sigma_v^k = \frac{w_v^k}{d_v^k + 1}, \tag{58}$$

where the identifier v breaks score ties. A vertex loses the repair competition when it is the lower-priority endpoint of at least one newly violated edge:

$$R^k = \left\{ v \in S^{k-1} : \exists \{u, v\} \in \mathcal{V}^k \text{ with } \eta_u^k > \eta_v^k \right\}. \tag{59}$$

The repaired set and initialization are

$$\widetilde{S}^k = S^{k-1} \backslash R^k, \tag{60}$$

$$a_v^k(0) = \mathbf{1}[v \in \widetilde{S}^k]. \tag{61}$$

Because η_v^k is a strict lexicographic order, every newly violated edge has a unique lower-priority endpoint; removing the union of these designated endpoints therefore destroys every newly violated edge. Hence $\widetilde{S}^k$ is independent in G^k. The better-response stage can add vertices that become locally unblocked after repair. Because all weights remain positive, a weight change alone does not make a conflict-free selected vertex leave in Stage I; locally inferior selections are instead replaced by the exchange stage.

6.2 Potential Drift

At epoch k, choose $\lambda_k > \bar{w}^k \geq w_{\max}^k$ and define

$$\Phi_k(\boldsymbol{a}) = \sum_{v \in V} w_v^k a_v - \lambda_k \sum_{\{u,v\} \in E^k} a_u a_v. \tag{62}$$

For a fixed action profile, the potential variation is bounded as follows.

Proposition 2 (Potential drift bound): *For any* $\boldsymbol{a} \in \{0,1\}^n$,

$$\begin{aligned} |\Phi_k(\boldsymbol{a}) - \Phi_{k-1}(\boldsymbol{a})| &\leq \delta_w^k + \lambda_k \delta_E^k + |\lambda_k - \lambda_{k-1}|\, C_{E^{k-1}}(\boldsymbol{a}) \\ &\leq \delta_w^k + \lambda_k \delta_E^k + |\lambda_k - \lambda_{k-1}|\, |E^{k-1}|. \end{aligned} \tag{63}$$

Proof: The vertex contribution is bounded by $\left\| \boldsymbol{w}^k - \boldsymbol{w}^{k-1} \right\|_1$. Add and subtract the old-edge conflict count under coefficient λ_k. Because $a_u a_v \in \{0,1\}$, changing the edge set contributes at most $\lambda_k |E^k \triangle E^{k-1}|$, while changing the penalty contributes exactly $|\lambda_k - \lambda_{k-1}| C_{E^{k-1}}(\boldsymbol{a})$. Finally, $C_{E^{k-1}}(\boldsymbol{a}) \leq |E^{k-1}|$. □

The bound explains why warm starting is useful in slowly varying systems: a small symmetric difference and small weight drift perturb the optimization landscape locally rather than replacing it entirely. It does not imply that the new optimum is identical to the old one; it quantifies the maximum change of the certificate at any fixed state. Warm starting is therefore expected to be advantageous when changes are sparse relative to the graph and the time between epochs is long enough to complete a repair/refinement episode. Under large global rewiring, abrupt weight re-ranking, or changes that arrive faster than optimization settles, the reusable part of the old set shrinks and a cold restart can become competitive.

6.3 Communication, Computation, and Storage

In Stage I, one logical round consists of an action broadcast followed by an eligibility/priority broadcast. Node v therefore receives one binary action, one eligibility flag, and one fixed-size priority tuple from each neighbor. Its local arithmetic cost is

$$\mathcal{C}_v^{\mathrm{BR}} = O(d_v), \tag{64}$$

and its stored one-hop state is

$$\mathcal{M}_v^{\mathrm{BR}} = O(d_v). \tag{65}$$

Across the network, one round uses

$$\mathcal{Q}^{\mathrm{BR}} = O(|E|) \tag{66}$$

fixed-size transmissions under standard neighbor broadcasting.

If $\bar{w}$ is not known a priori, synchronous max-consensus adds at most D_G preliminary rounds on a connected static graph; this calibration is excluded from the optimization-round curves but is now explicitly accounted for as initialization overhead. In Stage II, a selected blocker coordinates at most q shortlisted unselected neighbors. The candidates send their selected-neighbor identifiers and weights to that blocker, so pair construction uses radius-two information relayed over one-hop links rather than global topology. In the worst case, evaluating all shortlisted pairs costs

$$\mathcal{C}_v^{\mathrm{EX}} = O(d_v + q^2\Delta), \tag{67}$$

while cached candidate and conflict metadata require

$$\mathcal{M}_v^{\mathrm{EX}} = O(d_v + q\Delta + q^2). \tag{68}$$

Let $E_{\mathcal{P}}$ denote a proposal-conflict edge set and let H_{EX} be the maximum number of local contention micro-rounds within one outer exchange round. A conservative aggregate operation/message bound, using per-round worst-case values, is

$$\mathcal{T} = O\Big(T_{\mathrm{BR}}|E| + T_{\mathrm{EX}}\big(|S|q^2\Delta + H_{\mathrm{EX}}|E_{\mathcal{P}}|\big)\Big), \tag{69}$$

where T_{BR} and T_{EX} count outer optimization rounds and exclude optional max-consensus calibration. In the practical implementation, the mixed pair phase satisfies $T_{\mathrm{EX}}^{\mathrm{pair}} \le R = 25$; the subsequent singleton polishing is included in T_{EX}. Reported Stage-II counts treat a complete contention episode as one outer exchange round and do not count its micro-round message phases separately. The exchange messages are richer than the scalar Stage-I messages, but no node stores the full adjacency matrix or all vertex weights. For empirical accounting, one point-to-point transmission over one graph/proposal-conflict edge is counted as one message. Payload is reported in normalized scalar-field units: identifiers, weights, gains, flags, and priority fields each count as one field. Candidate relay packets additionally include the current selected-neighbor identifiers. This definition avoids hardware-specific byte assumptions while exposing the relative communication burden. Table 3 summarizes the information and repair properties of the compared methods, while Table 4 complements this comparison with empirical message and normalized-payload accounting.

Table 3: Qualitative comparison of MWIS baselines and proposed methods.

Method	Information per Round	Feasible Output	Monotone Certificate	Local Repair after Changes
DGA	One-hop weights and states	Yes	Greedy order only	Restart or rerun
RDA-10	Global random order; 10 restarts	Yes	Best-of-10 objective	Full restart
BP	Directed real messages	After decoding	Not guaranteed on loops	Message re-iteration
ADGA	Actions and priorities	Yes at equilibrium	Exact potential	Warm-start better response
ADGA-X	Actions, priorities, and proposals	Yes	Potential and exchange gain	Warm-start repair and exchange

Note: DGA: distributed greedy algorithm; RDA-10: ten-restart randomized greedy algorithm; BP: belief propagation; ADGA: adaptive distributed game algorithm; ADGA-X: exchange-refined ADGA.

Table 4: Empirical communication accounting over the 20 paired main instances.

Graph	Method	Messages/Instance	Payload Units/Instance
ER	DGA	44,044	88,088
	BP	300,660	300,660
	ADGA	93,202	233,006
	ADGA-X	99,827	275,053
BA	DGA	8499	16,997
	BP	89,460	89,460
	ADGA	20,278	50,694
	ADGA-X	23,891	82,339
RGG	DGA	8453	16,906
	BP	76,932	76,932
	ADGA	16,437	41,092
	ADGA-X	17,864	47,374

Note: Average contention micro-rounds per ADGA-X outer exchange round are 1.47 (ER), 1.39 (BA), and 1.30 (RGG). ER: Erdoš–Rényi; BA: Barabási–Albert-type; RGG: random geometric graph. RDA-10 is centralized and is therefore not assigned a peer-to-peer message count.

7 Experimental Evaluation

7.1 Algorithms, Graphs, and Metrics

The experiments use a common synchronous network emulator. Stage I accesses one-hop states and priorities, while Stage II relays bounded-radius proposal metadata through selected blockers. Four distributed or message-passing methods are compared with RDA-10 as a centralized multi-start quality baseline.

1. **DGA**: synchronous distributed greedy selection; in each round, unblocked local winners under $w_v/(d_v+1)$ are selected in parallel.
2. **RDA-10**: a simple centralized random-order multi-start baseline used as a weight-agnostic stress reference; the best feasible result from ten independently seeded random greedy orders is reported. Because its insertion order deliberately ignores weights and degrees, it should not be interpreted as a state-of-the-art centralized MWIS solver.
3. **BP**: a max-sum message-passing heuristic with 30 iterations, followed by feasible greedy decoding.
4. **ADGA**: Stage I of the proposed potential-game method.
5. **ADGA-X**: ADGA followed by at most $R = 25$ mixed outer exchange rounds with $q = 15$, then singleton-only polishing.

Three graph families are used. Erdős–Rényi (ER) graphs independently include each edge with probability p [22]. Barabási–Albert-type (BA) preferential-attachment graphs model hub-dominated scale-free connectivity [23]. Random geometric graphs (RGG) connect nodes within a spatial radius and approximate range-limited wireless or sensor interference. Unless otherwise stated, integer weights are sampled uniformly from $\{1,\dots,100\}$, so the known bound is $\bar{w} = 100$ and the common penalty is fixed at $\lambda = 1.05\,\bar{w} = 105$. For the empty or repaired feasible initializations used in all experiments, every tested $\lambda > w_{\max}$ gives the same Stage-I activation gains, while Stage II is evaluated only on feasible sets; changing λ within the admissible range therefore does not change the returned set in these tests. The 5% margin is a robustness margin for bound implementation rather than a quality-tuning parameter; arbitrary conflicting initial profiles may still follow different transient deactivation orders because the priority contains $|\Delta_v|$.

For a returned set S_A from algorithm A, the primary quality metric is

$$Q_A = W(S_A). \tag{70}$$

Feasibility is measured by

$$F_A = \mathbf{1}\left[\sum_{\{u,v\}\in E} \mathbf{1}[u \in S_A]\mathbf{1}[v \in S_A] = 0\right]. \tag{71}$$

For small instances with certified optimum $S^\star$, the optimality ratio is

$$R_A = \frac{W(S_A)}{W(S^\star)}. \tag{72}$$

The exchange improvement over ADGA is

$$I_X = 100 \times \frac{W(S_{\mathrm{ADGA-X}}) - W(S_{\mathrm{ADGA}})}{W(S_{\mathrm{ADGA}})}\%. \tag{73}$$

For a dynamic instance, the warm-start outer-round reduction is

$$D_T = 100 \times \left(1 - \frac{T_{\mathrm{warm}}}{T_{\mathrm{cold}}}\right)\%. \tag{74}$$

The main static experiments use 20 independent graph/weight realizations. The graph generator, weight sampler, priority tie breakers, exchange contention, and dynamic perturbations use distinct deterministic random streams; set-dependent iterations are explicitly sorted to avoid dependence on Python hash order. Exact tests use 20 ER graphs with $n = 22$ and $p = 0.16$, where an exact maximum-weight clique computation

on the complement graph certifies the MWIS weight. Scaling and density curves use four independent realizations per point and are retained only as descriptive trend plots. Inferential comparisons are instead based on the 20 paired main instances, for which a two-sided Wilcoxon signed-rank test and matched-pairs rank-biserial effect size are reported. All reported final sets passed the feasibility check; this is treated as a deterministic implementation check consistent with the feasibility guarantee, not as a discriminative performance metric. The archived run used Python 3.13.5, NetworkX 3.6.1, NumPy 2.3.5, and Matplotlib 3.10.8; these versions are pinned in the package together with the graph parameters, deterministic seeds, penalty, iteration limits, and exchange cap. Table 5 consolidates the principal experimental settings.

Table 5: Main experimental settings.

Item	Setting
Vertex weights	Independent discrete uniform on $\{1, \ldots, 100\}$
Penalty	$\bar{w} = 100$ and $\lambda = 1.05\,\bar{w} = 105$
Static trials	20 per main graph scenario
Scaling/density trials	4 per point
RDA-10	Best of 10 random orders
BP	30 message iterations and feasible decoding
Exchange candidate cap	$q = 15$
Mixed outer exchange-round cap	$R = 25$
Dynamic perturbation	5% edge toggles; 10% vertices with weight drift
Dynamic trials	20

Table 6 reports a small paired sensitivity study used to select the practical caps on 12 ER instances with $n = 500$ and $p = 0.04$. Increasing the candidate/round budgets improves weight rapidly from $(q, R) = (5, 10)$ to $(15, 25)$, after which the additional gain is small relative to the extra proposal traffic. This supports $q = 15$ and $R = 25$ as a quality–overhead compromise rather than a theoretically privileged choice.

Table 6: Sensitivity of the practical exchange caps (12 paired ER instances).

q	R	Final Weight	Outer Rounds	Micro-Rounds/Exchange
5	10	5725.25 ± 316.01	7.67 ± 1.23	1.02 ± 0.36
10	15	5990.00 ± 270.49	9.08 ± 0.79	1.22 ± 0.23
15	25	6108.67 ± 263.58	10.50 ± 2.07	1.30 ± 0.11
25	40	6137.75 ± 251.31	10.33 ± 2.15	1.39 ± 0.21

7.2 Static Performance on Three Graph Families

Table 7 reports mean total weight, standard deviation, and round count from the 20 paired validation instances. On ER graphs, ADGA-X obtains 6016.30, exceeding ADGA by

$$I_X^{\mathrm{ER}} = 11.61\%. \tag{75}$$

Table 7: Static performance: mean ± standard deviation over 20 paired trials.

Algorithm	ER ($n = 500, p = 0.04$)	BA ($n = 500, m = 3$)	RGG ($n = 250, r = 0.12$)
DGA	5676.15 ± 163.93	13,697.15 ± 477.64	3281.10 ± 146.78
RDA-10	4248.70 ± 212.45	11,278.90 ± 462.92	2372.60 ± 189.39
BP	5444.85 ± 193.76	**14,188.65 ± 503.28**	3193.85 ± 170.26
ADGA	5390.65 ± 171.31	13,039.85 ± 746.25	3165.90 ± 153.14
ADGA-X	**6016.30 ± 153.86**	14,107.45 ± 530.87	**3478.15 ± 143.64**
ADGA rounds	4.65 ± 0.59	3.40 ± 0.60	3.20 ± 0.62
ADGA-X outer rounds	9.20 ± 1.96	6.35 ± 1.18	5.65 ± 0.93

Note: Bold values indicate the highest mean selected weight among the compared algorithms for each graph family.

On BA graphs, it improves the base equilibrium by

$$I_X^{\mathrm{BA}} = 8.19\%, \tag{76}$$

and on RGGs the gain is

$$I_X^{\mathrm{RGG}} = 9.86\%. \tag{77}$$

These gains show why unilateral equilibrium alone is insufficient: pair exchanges can replace a locally blocking configuration while preserving feasibility.

BP gives the largest mean on the BA scenario, whereas ADGA-X is strongest on ER and RGG graphs. Because every algorithm is evaluated on the same 20 instances, paired Wilcoxon signed-rank tests are appropriate. For ADGA-X vs. BP, the two-sided p-values are 1.91×10^{-6} (ER), 2.67×10^{-5} (BA), and 1.91×10^{-6} (RGG), with matched-pairs rank-biserial effects 1.00, −0.943, and 1.00, respectively. Thus ADGA-X has a clear paired advantage on ER/RGG, while BP has a clear paired advantage on this BA configuration; no topology-independent dominance is claimed. All algorithms returned feasible decoded/constructed sets; feasibility is therefore reported as a verified property rather than a ranking metric.

7.3 Certified Small-Instance Quality

Table 8 compares each heuristic with the exact optimum on 20 small ER instances. BP attains the largest mean ratio, 0.9938, while ADGA-X reaches 0.9899; their difference is small relative to the trial-to-trial variability. ADGA-X substantially improves the base ADGA mean from 0.9606, confirming the practical value of the exchange stage.

Table 8: Exact-optimum ratios on ER graphs ($n = 22$, $p = 0.16$, 20 trials).

Algorithm	Mean $W(S_A)/W(S^\star)$	Standard Deviation
DGA	0.9789	0.0328
RDA-10	0.9293	0.0589
BP	**0.9938**	0.0165
ADGA	0.9606	0.0557
ADGA-X	0.9899	0.0237

Note: Bold value indicates the highest mean exact-optimum ratio among the compared algorithms.

The empirical ratio is not a proof of near-optimality for large graphs. The general guarantee remains (51); exact comparison is limited to instances where the optimum can be certified.

7.4 Scalability and Density Sensitivity

Fig. 1 shows the selected weight for ER graphs with $p = 0.03$ as n increases from 100 to 1000. Error bars represent one standard deviation over four realizations. ADGA-X maintains competitive or leading weight throughout the tested range. The advantage becomes visible at $n = 500$ and above, where more local exchange opportunities exist.

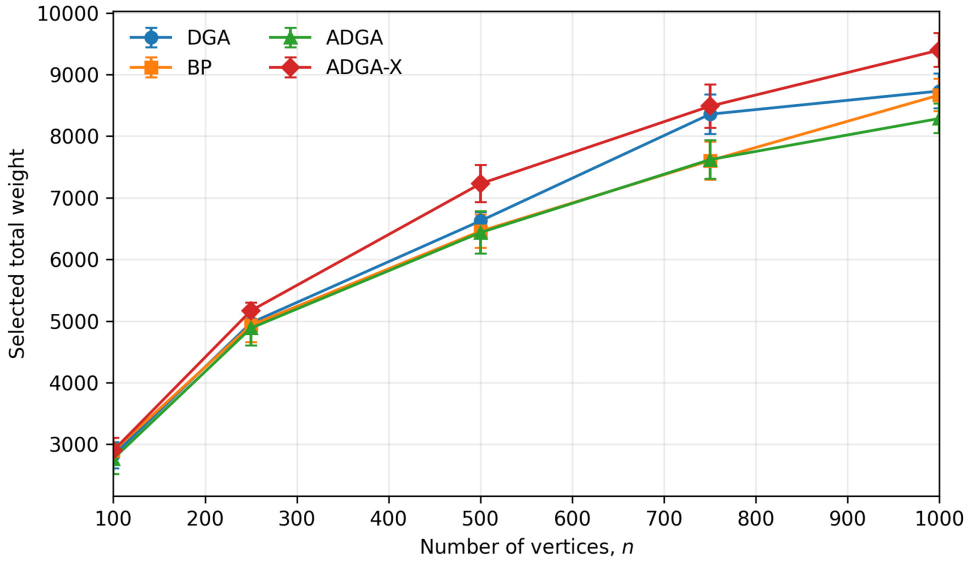


Figure 1: Selected total weight vs. network size on Erdoś–Rényi (ER) graphs with $p = 0.03$; mean and standard deviation over four realizations. DGA: distributed greedy algorithm; BP: belief propagation; ADGA: adaptive distributed game algorithm; ADGA-X: exchange-refined adaptive distributed game algorithm.

Fig. 2 reports outer update rounds or iterations. Synchronous DGA converges in 2.00–5.00 local-winner rounds, BP is fixed at 30 message iterations, and ADGA converges in 2.00–5.25 parallel better-response rounds; ADGA-X uses additional outer exchange rounds. These logical counts are not equal wall-clock or message-phase costs: one BP iteration transmits directed real messages, while one ADGA-X outer exchange round may contain several proposal-contention micro-rounds before spatially compatible exchanges are executed. Across the 20 paired main settings, however, the average contention depth is only 1.47, 1.39, and 1.30 micro-rounds per outer exchange round for ER, BA, and RGG graphs, respectively; Table 4 reports the corresponding message and normalized payload totals.

Fig. 3 varies p from 0.02 to 0.10 for $n = 400$. Total weight decreases as the graph becomes denser because each selected vertex excludes more candidates. ADGA-X gives the largest mean at every tested density in these runs, although its margin varies and the four-realization curves should be interpreted descriptively rather than as significance tests.

Dense graphs reduce the availability of internally independent pair proposals, so the relative benefit of two-vertex exchanges can still weaken.

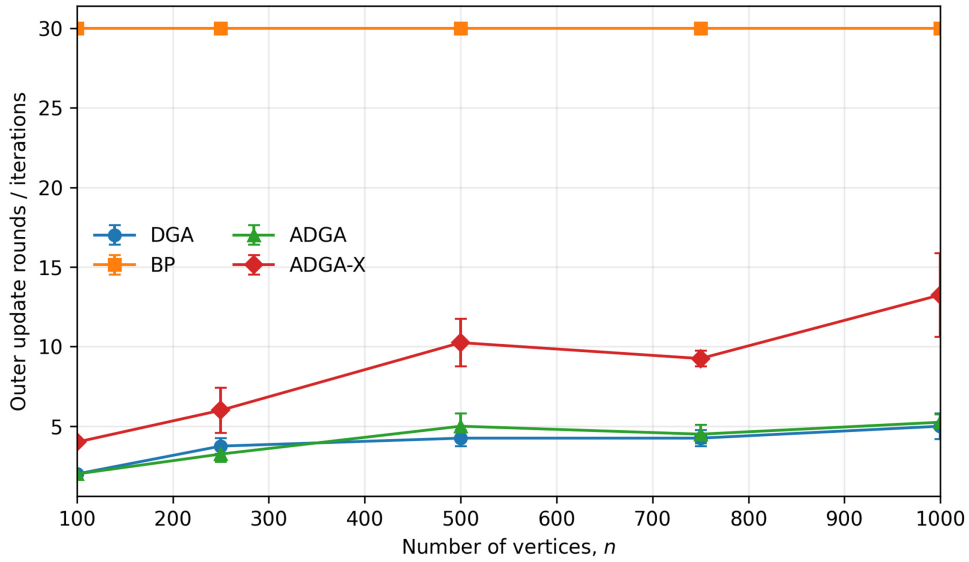


Figure 2: Outer update rounds or iterations vs. network size on ER graphs with $p = 0.03$; Stage-II contention micro-rounds are not counted separately.

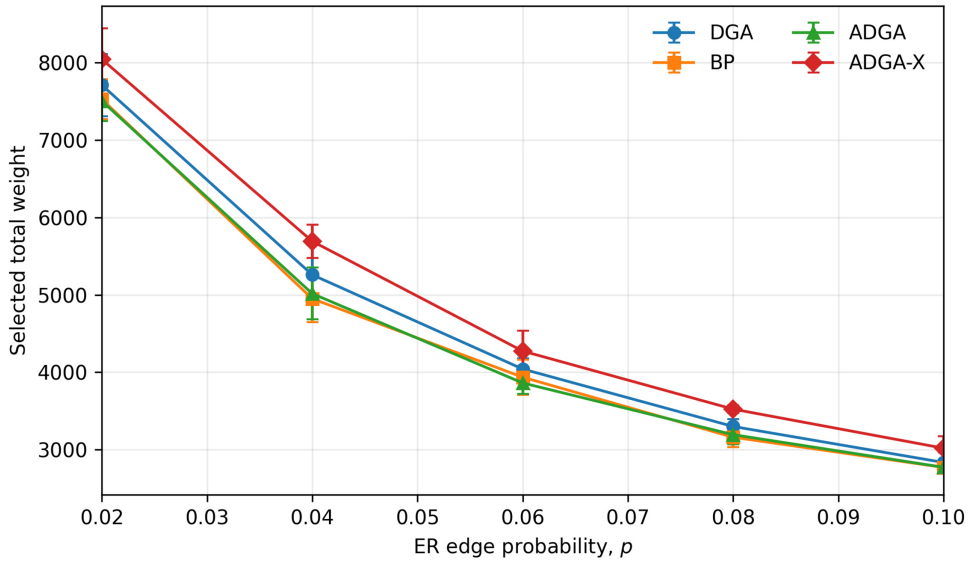


Figure 3: Selected total weight vs. ER edge probability for $n = 400$; mean and standard deviation over four realizations.

7.5 Dynamic Warm-Start Evaluation

The dynamic experiment begins with an ER graph having $n = 300$ and $p = 0.04$. After convergence, approximately 5% of its edges are toggled, and 10% of vertices receive nonzero bounded weight perturbations. The warm method first applies the explicit repair in (59)–(61) and then reruns ADGA-X, whereas the cold method starts from the empty state. Both variants use the same priority and contention seeds, so the comparison changes the initialization rather than intentionally changing the tie-breaking policy. One repair pass is included in the warm recovery outer-round count. Before repair, the old solution contains on average 5.30 ± 2.99 newly violated edges, and repair removes 5.05 ± 2.70 vertices. Table 9 reports 20 trials.

Table 9: Dynamic recovery after topology and weight changes (20 paired trials).

Metric	Warm Start	Cold Start
Recovery outer rounds	4.45 ± 2.28	8.40 ± 1.47
Final selected weight	4751.70 ± 262.48	4743.05 ± 244.45
Warm/cold weight ratio	1.0020 ± 0.0274	

The ratio is computed trial by trial before taking its mean and standard deviation; it ranges from 0.9590 to 1.0679, which makes the speed–quality tradeoff explicit rather than implying that warm starting must improve the final local optimum.

Using (74) and the reported mean round counts, the reduction is

$$D_T = 100\left(1 - \frac{4.45}{8.40}\right) = 47.02\%. \tag{78}$$

The warm-start and cold-start final means are statistically close relative to their trial-to-trial dispersion, and individual trials can favor either initialization. This path dependence confirms that warm starting does not guarantee a higher-weight local optimum. Its robust advantage in this experiment is faster recovery because most unaffected local decisions are retained.

Fig. 4 complements the aggregate table with a process-level view: local repair supplies a feasible initial point and subsequent accepted moves monotonically raise the certificate.

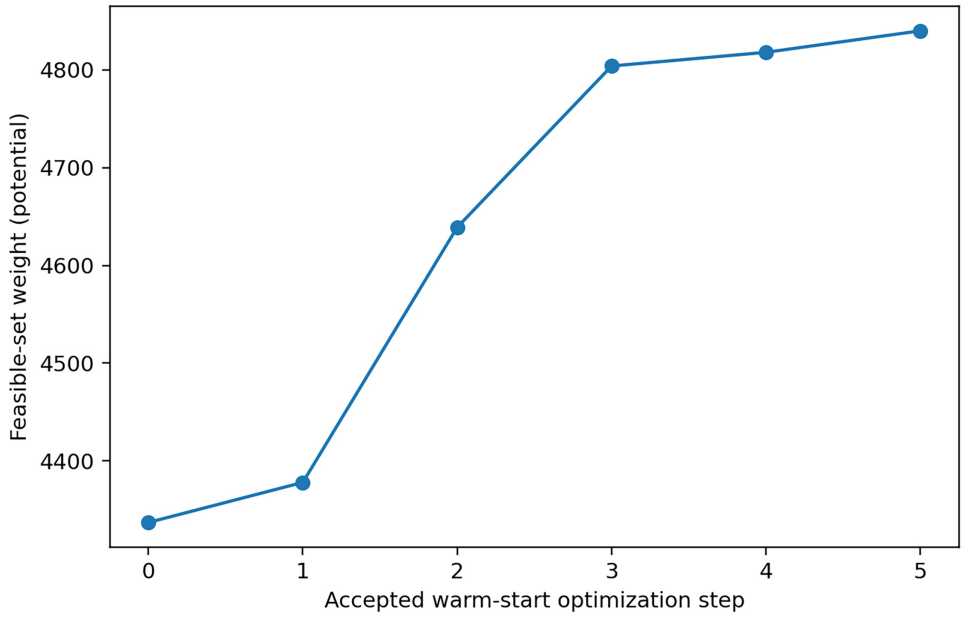


Figure 4: Representative warm-start trajectory. Because repair preserves feasibility, the plotted feasible-set weight equals the exact potential during the accepted optimization steps; each accepted Stage-I or exchange step is nondecreasing. The horizontal axis is a unitless accepted-step count, and the vertical axis is the aggregate vertex-weight score in the dimensionless utility units used for w_v.

7.6 Ablation, Interpretation, and Limitations

The clearest ablation is ADGA vs. ADGA-X. Both use the same penalty game and parallel better-response stage; only ADGA-X performs exchanges. The 8.19%–11.61% gains in (75)–(77) therefore isolate the value of exchange refinement. The feasibility rate remains unchanged because every accepted exchange explicitly removes the full selected conflict set.

Three limitations should be emphasized. First, the $1/\Delta$ guarantee in Theorem 5 is worst-case and can be loose for dense weighted graphs. Second, candidate truncation $q = 15$ and the mixed-round budget $R = 25$ sacrifice full two-vertex stability for predictable runtime, although singleton polishing preserves the stated approximation guarantee; Table 6 quantifies the associated quality–communication tradeoff. Third, the experiments use synthetic conflict graphs rather than a trace-driven production network; consequently, the numerical gains should not be interpreted as deployment-specific guarantees. The graph families nevertheless cover independent-edge, hub-dominated, and geometric structures, and the supplied scripts enable direct extension to application traces.

The method also has a clear relationship to learning-assisted systems. A learning model may predict weights, link risks, or task values; ADGA-X can then enforce conflict feasibility and perform decentralized optimization on the predicted graph. Conversely, ADGA-X itself does not require offline training, a centralized critic, or a replay buffer. Compared with GNN-based MWIS scheduling [20], this removes training-data and distribution-shift requirements but forgoes learned global structure; compared with ADMM-style distributed optimization [21], it works directly with binary decisions and exact conflict feasibility but incurs combinatorial proposal coordination. This makes it suitable for safety-critical or rapidly changing settings where a deterministic local improvement certificate is preferred.

8 Conclusion

This paper developed ADGA-X, a distributed potential-game method with local exchanges for static and dynamic MWIS problems. The penalty-dominant utility induces an exact potential, and the threshold $\lambda > \bar{w} \geq w_{\max}$, which is implementable from a system ceiling or finite max-consensus, guarantees feasible maximal equilibria and excludes infeasible global maximizers. A nonadjacent priority schedule permits parallel strict better responses while retaining additive potential improvement. Exhaustive one- and two-vertex exchanges then repair poor unilateral equilibria and terminate at full exchange stability; the candidate- and round-capped implementation uses singleton polishing to retain finite convergence, singleton stability, and the resulting degree-dependent guarantee. For changing graphs, warm-start repair bounds newly created conflicts and reuses unaffected decisions.

The practical significance is that a node needs only one-hop state exchange in Stage I and bounded-radius proposal relay in Stage II, while all accepted states remain conflict-free. The revised communication accounting shows that the exchange layer adds moderate traffic relative to ADGA in the tested graphs, and the cap study exposes the quality–overhead tradeoff rather than hiding it. Paired tests further show that topology matters: ADGA-X is significantly stronger than the tested BP decoder on ER/RGG instances, whereas BP is stronger on the BA setting. Experiments showed that exchange refinement increased the base potential-game weight by approximately 8.2%–11.6% on three graph families, attained a mean exact-optimum ratio of 0.990 on small instances, and reduced dynamic recovery outer rounds by about 47.0%.

The main limitations are the worst-case $1/\Delta$ guarantee, the loss of full pair stability under practical truncation, the extra two-hop/proposal communication in dense graphs, and the assumption that a graph snapshot is stable during one optimization episode. Very frequent global changes can erode the advantage of warm starting, and the current evaluation remains synthetic rather than trace-driven. Future work will

therefore study asynchronous packet delays, adaptive candidate budgets, tighter graph-class-dependent bounds, and trace-based integration with learned weight or topology predictors.

Acknowledgement: Not applicable.

Funding Statement: This research was funded by the National Natural Science Foundation of China, Grant No. 62103178.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Changlei Zhan and Yong Ding; methodology and formal analysis: Changlei Zhan and Yong Ding; software, experiments, and validation: Changlei Zhan; analysis and interpretation of results: Changlei Zhan and Yong Ding; draft manuscript preparation: Changlei Zhan; manuscript revision and supervision: Yong Ding. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The authors confirm that the data supporting the findings of this study are available within the article and its supplementary materials. The revision validation script, deterministic seeds, and aggregated statistical results are included with the source package.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

Abbr.	Definition
MWIS	Maximum-weight independent set
MIS	Maximal independent set
ADGA-X	Exchange-refined adaptive distributed game algorithm
DGA	Distributed greedy algorithm
RDA-10	Ten-restart randomized greedy algorithm
BP	Belief propagation
ADGA	Adaptive distributed game algorithm
ER	Erdoš–Rényi
BA	Barabási–Albert-type
RGG	Random geometric graph
GNN	Graph neural network
ADMM	Alternating direction method of multipliers
RFID	Radio-frequency identification

References

1. Tarjan RE, Trojanowski AE. Finding a maximum independent set. SIAM J Comput. 1977;6(3):537–46. doi:10.1137/0206038.
2. Sanghavi S, Shah D, Willsky AS. Message passing for maximum weight independent set. IEEE Trans Inform Theory. 2009;55(11):4822–34. doi:10.1109/tit.2009.2030448.
3. Tan Z, Chen B, Li R, Yang L, Xu X. Efficient cache retrieving scheme in VANET via online reinforcement learning. IEEE Trans Veh Technol. 2025;74(6):9506–19. doi:10.1109/tvt.2025.3532836.
4. Verma S, Rodrigues TK, Kawamoto Y, Fouda MM, Kato N. A survey on multi-AP coordination approaches over emerging WLANs: future directions and open challenges. IEEE Commun Surv Tutor. 2024;26(2):858–89. doi:10.1109/comst.2023.3344167.
5. Zhang S, Bao S, Chi K, Yu K, Mumtaz S. DRL-based computation rate maximization for wireless powered multi-AP edge computing. IEEE Trans Commun. 2024;72(2):1105–18. doi:10.1109/tcomm.2023.3325905.

6. Hsu BY, Yang CH, Lu CH, Chang MY, Yeh LY, Shen CY. Diversifying graph augmentation for learning to solve graph optimization problems. IEEE Trans Knowl Data Eng. 2025;37(12):7060–75. doi:10.1109/tkde.2025.3611663.
7. Gadouleau M, Kutner DC. Generalising the maximum independent set algorithm via Boolean networks. Inf Comput. 2025;303:105266. doi:10.1016/j.ic.2025.105266.
8. Ebadi S, Keesling A, Cain M, Wang TT, Levine H, Bluvstein D, et al. Quantum optimization of maximum independent set using Rydberg atom arrays. Science. 2022;376(6598):1209–15. doi:10.1126/science.abo6587.
9. Zhu E, Jiang F, Liu C, Xu J. Partition independent set and reduction-based approach for partition coloring problem. IEEE Trans Cybern. 2022;52(6):4960–9. doi:10.1109/tcyb.2020.3025819.
10. Lamm S, Schulz C, Strash D, Williger R, Zhang H. Exactly solving the maximum weight independent set problem on large real-world graphs. In: 2019 Proceedings of the Twenty-First Workshop on Algorithm Engineering and Experiments (ALENEX). Philadelphia, PA, USA: Society for Industrial and Applied Mathematics; 2019. p. 144–58. doi:10.1137/1.9781611975499.12.
11. Luby M. A simple parallel algorithm for the maximal independent set problem. SIAM J Comput. 1986;15(4):1036–53. doi:10.1137/0215074.
12. Öztemiz F. A greedy approach to solve maximum independent set problem: differential Malatya independent set algorithm. Eng Sci Technol Int J. 2025;63:101995. doi:10.1016/j.jestch.2025.101995.
13. Andrade DV, Resende MGC, Werneck RF. Fast local search for the maximum independent set problem. J Heuristics. 2012;18(4):525–47. doi:10.1007/s10732-012-9196-4.
14. Lamm S, Sanders P, Schulz C, Strash D, Werneck RF. Finding near-optimal independent sets at scale. J Heuristics. 2017;23(4):207–29. doi:10.1007/s10732-017-9337-x.
15. Jin Y, Hao JK. General swap-based multiple neighborhood tabu search for the maximum independent set problem. Eng Appl Artif Intell. 2015;37:20–33. doi:10.1016/j.engappai.2014.08.007.
16. Monderer D, Shapley LS. Potential games. Games Econ Behav. 1996;14(1):124–43. doi:10.1006/game.1996.0044.
17. Chen J, Zhou R. Yet another self-stabilizing minimum vertex cover of a network with stochastic stability. IEEE Trans Control Netw Syst. 2024;11(4):2226–38. doi:10.1109/tcns.2024.3395725.
18. He H, Duan H, Yuan W, Zeng Z. A potential game approach to target assignment in heterogeneous manned/unmanned aerial team with incomplete information. IEEE Trans Circuits Syst II. 2024;71(12):4894–8. doi:10.1109/tcsii.2023.3287160.
19. Zeng L, Sa P, Han Q, Ye L, Yang L, Zhang C, et al. DRG-DCC: a driving risk gaming based distributed congestion control method for C-V2X technology. CMC. 2025;83(2):2059–86. doi:10.32604/cmc.2025.060392.
20. Zhao Z, Verma G, Rao C, Swami A, Segarra S. Distributed scheduling using graph neural networks. In: Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); 2021 Jun 6–11; Toronto, ON, Canada. p. 4720–4. doi:10.1109/icassp39728.2021.9414098.
21. Boyd S, Parikh N, Chu E, Peleato B, Eckstein J. Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends Mach Learn. 2011;3(1):1–122. doi:10.1561/2200000016.
22. Erdős P, Rényi A. On random graphs I. Publ Math Debrecen. 2011;6:290–7.
23. Bollobás B, Riordan O, Spencer J, Tusnády G. The degree sequence of a scale-free random graph process. Random Struct Algorithms. 2001;18(3):279–90. doi:10.1002/rsa.1009.