

**МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ, ЧИСЛЕННЫЕ МЕТОДЫ И КОМПЛЕКСЫ  
ПРОГРАММ/MATHEMATICAL MODELING, NUMERICAL METHODS AND PROGRAM COMPLEXES**

**DOI: <https://doi.org/10.60797/IRJ.2025.158.94>**

**A NOVEL FRAMEWORK FOR MULTI-AGENT MULTI-CRITERIA DECISION-MAKING: MATHEMATICAL  
MODELING AND RASTRIGIN-BASED BENCHMARKING**

Research article

**Hammoud A.<sup>1,\*</sup>**

<sup>1</sup> ORCID : 0009-0004-6063-0327;

<sup>1</sup> Kuban State Agrarian University, Krasnodar, Russian Federation

\* Corresponding author (ali-hammoud[at]mail.ru)

**Abstract**

Multi-agent systems (MAS) are increasingly adopted in various domains, necessitating efficient decision-making strategies that balance individual agent preferences with overarching system objectives. This paper presents a novel framework for Multi-Agent Multi-Criteria Decision-Making (MAS-MCDM), which extends any MCDM algorithm to address the challenges of scalability, resource constraints, and inter-agent interactions. We propose four distinct decision-making approaches:

1. Full Enumeration, which guarantees the global optimum but is computationally prohibitive for large-scale problems.
2. Independent Agent Decomposition, which enhances scalability but neglects inter-agent synergies.
3. Iterative Refinement, where agents adjust decisions dynamically based on system-level feedback to satisfy constraints.
4. Iterative Refinement with Inter-Agent Interactions, which integrates cooperation and competition dynamics through an interaction matrix.

To validate the framework, we adapt the Rastrigin function — traditionally used for single-agent global optimization — into a discrete MAS-MCDM benchmark. By transforming the continuous optimization problem into a decision-making task, we examine how constraints and interactions influence solutions, shifting them away from the global minimum at ( $x=0$ ). The experiments demonstrate that iterative refinement approaches effectively navigate resource limitations, while interaction-aware models enable the emergence of cooperative and competitive behaviors. The results highlight trade-offs between optimality, scalability, and inter-agent coordination, providing insights into designing robust multi-agent systems for multi-criteria decision-making (MAS-MCDM) strategies in real-world applications. Future directions include refining interaction models, integrating reinforcement learning, and extending applications to autonomous and resource-constrained systems.

**Keywords:** Multi-Agent Systems (MAS), Multi-Criteria Decision-Making (MCDM), optimization framework, inter-agent interactions, benchmarking with Rastrigin function.

**НОВАЯ МЕТОДОЛОГИЯ ДЛЯ МУЛЬТИАГЕНТНОГО МНОГОКРИТЕРИАЛЬНОГО ПРИНЯТИЯ  
РЕШЕНИЙ: МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ И БЕНЧМАРКИНГ НА ОСНОВЕ ФУНКЦИИ  
РАСТРИГИНА**

Научная статья

**Хаммуд А.<sup>1,\*</sup>**

<sup>1</sup> ORCID : 0009-0004-6063-0327;

<sup>1</sup> Кубанский государственный аграрный университет, Краснодар, Российская Федерация

\* Корреспондирующий автор (ali-hammoud[at]mail.ru)

**Аннотация**

Мультиагентные системы (MAS) находят всё более широкое применение в различных областях, что требует разработки эффективных стратегий принятия решений, балансирующих предпочтения отдельных агентов с глобальными целями системы. В данной работе представлена новая методология для мультиагентного многокритериального принятия решений (MAS-MCDM), которая расширяет любой алгоритм MCDM для решения задач масштабируемости, ограничений ресурсов и межагентных взаимодействий. Мы предлагаем четыре подхода:

1. Полный перебор — гарантирует глобальный оптимум, но вычислительно неэффективен для крупномасштабных задач.
2. Декомпозиция независимых агентов — повышает масштабируемость, но игнорирует синергию между агентами.
3. Итеративное уточнение — агенты динамически корректируют решения на основе системной обратной связи для соблюдения ограничений.
4. Итеративное уточнение с межагентными взаимодействиями — интегрирует динамику кооперации и конкуренции через матрицу взаимодействий.

Для валидации методологии мы адаптировали функцию Растригина (традиционно используемую для однократной глобальной оптимизации) в дискретный MAS-MCDM бенчмарк. Преобразуя задачу непрерывной оптимизации в задачу принятия решений, мы исследуем влияние ограничений и взаимодействий на смещение решений относительно глобального минимума ( $x=0$ ). Эксперименты показали, что итеративные методы эффективно справляются с ограничениями ресурсов, а модели с учётом взаимодействий позволяют выявить кооперативное и конкурентное поведение агентов. Результаты демонстрируют компромиссы между оптимальностью, масштабируемостью и координацией агентов, что важно для проектирования устойчивых мультиагентных систем в реальных приложениях.

Перспективные направления включают уточнение моделей взаимодействий, интеграцию обучения с подкреплением и применение методологии в автономных и ресурсно-ограниченных системах.

**Ключевые слова:** мультиагентные системы (MAS), многокритериальное принятие решений (MCDM), оптимизационная методология, межагентные взаимодействия, бенчмаркинг с использованием функции Растригина.

## Introduction

Multi-Criteria Decision-Making (MCDM) serves as an essential methodology for evaluating, selecting, or ranking alternatives in complex scenarios where conflicting objectives and diverse criteria must be systematically balanced [1]. From infrastructure planning to autonomous systems, MCDM offers structured frameworks for synthesizing quantitative and qualitative factors, enabling stakeholders to navigate trade-offs and align decisions with strategic objectives [2].

As decision-making complexity escalates in group settings, Group Decision-Making (GDM) extends the MCDM paradigm to scenarios where a group of decision-makers (DMs) collaborates to reach a consensus [3]. In such collaborative environments, MCDM algorithms play a critical role by aggregating heterogeneous preferences and information — expressed through diverse preference relations (e.g., preference orderings, utility functions, multiplicative preference relations, and fuzzy preference relations), or varying data formats (e.g., including crisp values, interval numbers, fuzzy numbers, and linguistic data) — into a unified decision [4]. However, traditional GDM approaches often suffer from significant limitations: they may ignore the complex interrelationships and interactions among experts, struggle with scalability as group size increases and face challenges such as dominant influences, conflict resolution, and time-intensive consensus processes, leading to inconsistent opinions, biases from heterogeneous information representations, and difficulties in achieving true consensus.

In contrast, Multi-Agent Systems (MAS) provide a decentralized framework in which autonomous agents make independent decisions while interacting within a shared environment [5]. Unlike GDM's collaborative consensus-seeking approach, MAS empowers each agent to achieve both individual and system-level objectives, making decisions based on local information while also accounting for interdependencies, such as shared resource constraints and competitive or cooperative interactions [6]. Current MAS decision-making paradigms employ diverse approaches to address the challenges associated with coordination, learning, adaptability, and optimization among autonomous agents. These approaches include rule-based systems (primarily fuzzy logic), game-theoretic methods, evolutionary algorithms, multi-agent reinforcement learning (MARL), and large language models (LLMs) [7]. Although many MCDM methods have been successfully applied in both single-agent or group decision-making contexts, their extension to multi-agent systems presents unique challenges. In particular, the limitations of conventional GDM motivate the exploration of more robust, multi-layered decision frameworks that integrate MCDM into MAS.

Notably, there is no universally superior MCDM technique or algorithm, as the appropriateness of a method depends on the specific decision situation. While almost all methodologies in MCDM share common steps like problem organization and decision matrix construction, they differ in how they synthesize information [8]. While our prior work introduced the TriMetric Fusion (TMF) algorithm for single-agent MCDM [9], the present framework is designed to accommodate any MCDM technique, including TOPSIS, AHP [10], or any other suitable method, depending on problem-specific requirements.

The proposed Multi-Agent Multi-Criteria Decision-Making (MAS-MCDM) framework is a novel approach that addresses the challenges of scalability and global optimality in decentralized decision-making systems. It enables agents to rank alternatives based on local criteria while simultaneously negotiating system-wide constraints. Four models are introduced:

First, a full enumeration strategy exhaustively evaluates all combinations to guarantee global optimality, although it is computationally infeasible for large-scale problems.

Second, a decomposition approach enables independent agent evaluations, improving scalability at the potential cost of global optimality.

Third, an iterative refinement mechanism enables agents to adjust their decisions based on system-level feedback, thereby enhancing feasibility under resource constraints.

Finally, an extended iterative framework incorporates explicit inter-agent interaction terms to model cooperative synergies and competitive conflicts among agents.

To validate the framework, we adapt the Rastrigin function [11] — a benchmark in continuous optimization—to multi-agent decision-making by discretizing its input space. This adaptation transforms the problem into one where agents select options from predefined sets, linking the Rastrigin function's local minima landscape to multi-agent decision dynamics. This approach tests the framework's ability to manage resource limitations and inter-agent interactions.

The remainder of the paper is organized as follows. Section 2 details the methodology, including mathematical models and workflows for each approach. Section 3 validates the framework experimentally using the discretized Rastrigin function, with a focus on resource constraints and inter-agent dynamics. Section 4 concludes with a discussion of strengths, limitations, and future research directions.

## Methodology

### 2.1. Problem Setup

We define the following parameters:

$N$  : Number of agents (indexed by  $i \in 1, \dots, N$ ).

$M$  : Number of options for each agent (indexed by  $j \in 1, \dots, M$ ).

$x_{ij} \in \{0, 1\}$  : Binary decision variable.

For all following approaches, the decision variable  $x_{ij}$  is defined as:

$$x_{ij} = \begin{cases} 1, & \text{if agent } i \text{ selects option } j, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

If there are  $N$  agents, each with  $M$  options, the total number of binary decision variables is  $N \times M$ .

$p_{ij}$  : Preference score of agent  $i$  for option  $j$ .

The preference scores represent the normalized ranks of options that are calculated and ranked by the TMF algorithm. Based on the criteria for each agent, the TMF gives the value of the preference for each option.

$f_{ij}$  : Contribution of option  $j$  for agent  $i$  to system-wide goals (e.g., resource usage, efficiency, fairness). This parameter reflects the meaning of MCDM in the system level.

$g_{ijk}$  : Resource usage of agent  $i$  choosing option  $j$  under resource-constraint  $k$ .

$G_k$  : Resource-limit for constraint  $k$ .

## 2.2. Approach 1: Full Enumeration of $M^N$ Solutions

This approach systematically evaluates all possible combinations of options for  $N$  agents, each with  $M$  options. The total number of combinations is  $M^N$ . It is the most exhaustive method and guarantees the discovery of the globally optimal solution. However, its computational complexity grows exponentially with  $M$  and  $N$ , making it practical only for small problem sizes.

### 2.2.1. Mathematical Model

Agent-level Objective ( $Z_1$ ):

$$Z_1 = \sum_{i=1}^N \sum_{j=1}^M p_{ij} x_{ij} \quad (2)$$

System-Level Objective ( $Z_2$ ):

$$Z_2 = \sum_{i=1}^N \sum_{j=1}^M f_{ij} x_{ij} \quad (3)$$

Objective Function:

Combine agent objectives and system-level objectives:

$$Z = \alpha \sum_{i=1}^N \sum_{j=1}^M p_{ij} x_{ij} + \beta \sum_{i=1}^N \sum_{j=1}^M f_{ij} x_{ij} \quad (4)$$

where:

$p_{ij}, f_{ij}$  : normalized to the same scale.

$\alpha, \beta$  : Weights for agent and system objectives, respectively.

Ensure  $\alpha + \beta = 1$ .

Constraints:

Single Option Per Agent: Each agent selects exactly one option:

$$\sum_{j=1}^M x_{ij} = 1, \quad \forall i \in 1, \dots, N. \quad (5)$$

System Constraints: The system imposes constraints such as:

Resource limits:

$$\sum_{i=1}^N \sum_{j=1}^M g_{ijk} x_{ij} \leq G_k, \quad \forall k \in 1, \dots, K \quad (6)$$

Set of Solutions: Each solution  $S_k$  is a combination of options:

$$S_{\text{options}} = (1, j_1), (2, j_2), \dots, (N, j_N) \mid j_i \in 1, 2, \dots, M \text{ for each } i = 1, \dots, N \quad (7)$$

Optimal Solution: The optimal solution is:

$$S^* = \arg \max_{\text{options} \in \{1, 2, \dots, N\}} Z(S_{\text{options}}) \quad (8)$$

### 2.2.2. Strengths and Limitations of the Approach

This approach guarantees the globally optimal solution by systematically evaluating all  $M^N$  possible combinations, making it the most exhaustive and mathematically rigorous method for multi-agent decision-making. However, its exponential complexity makes it infeasible for large-scale problems, as the number of possible solutions grows rapidly with the number of agents and options. To address this, branch-and-bound techniques, or Mixed Integer Programming (MIP) solvers can be employed to reduce computational costs while still approximating optimal solutions. Additionally, fairness concerns arise because the optimization is based on a weighted sum, which may disproportionately disadvantage some agents. Furthermore, this approach assumes centralized decision-making, where a single entity has full knowledge of all agents' preferences and constraints, which is unrealistic in many real-world scenarios. While this approach provides theoretical guarantees of optimality, its practical application is limited, and more scalable decision-making frameworks should be considered for real-world implementations.

## 2.3. Approach 2: Decomposition into Agent Contributions $M \cdot N$

This approach reduces the computational complexity of multi-agent decision-making by evaluating each agent's decision independently and approximating the system-level objectives as the sum of individual agent contributions. While it may not guarantee the global optimum and does not consider the existence of the resources constraints like the full enumeration approach, it significantly improves scalability and remains practical for larger systems without constraints.

### 2.3.1. Mathematical Model

#### 2.3.1.1. Agent-Level Objective

Each agent independently evaluates its options:

$$Z_i = \sum_{j=1}^M (\alpha \cdot p_{ij} + \beta \cdot f_{ij}) x_{ij} \quad (9)$$

where:

$\alpha, \beta$  : Weights for agent-level preference and system contribution, and  $\alpha + \beta = 1$ .

### 2.3.1.2. System-Level Objective

Approximate the system-level objective as the sum of agent-level contributions:

$$Z_{\text{system}} = \sum_{i=1}^N \sum_{j=1}^M (\alpha \cdot p_{ij} + \beta \cdot f_{ij}) x_{ij} \quad (10)$$

### 2.3.1.3. Constraints

Each agent must select exactly one option:

$$\sum_{j=1}^M x_{ij} = 1, \quad \forall i \in 1, 2, \dots, N. \quad (11)$$

### 2.3.1.4. Optimal Solution

The optimal solution selects the best option for each agent based on their individual objectives and system-level contribution:

$$S^* = (i, j_i^*) \mid j_i^* = \arg \max_{j \in 1, 2, \dots, M} Z_i(x_{ij}), \forall i \in 1, 2, \dots, N \quad (12)$$

### 2.3.2. Step-by-Step workflow

Step 1: Define Agent Contributions

For each agent  $i$ , define its contribution  $f_{ij}$  which is the system-level evaluation of agent  $i$ 's choice of option  $j$ .

For example:  $f_{ij}$  could represent the energy saved, the load balance improved, or the efficiency gained by agent  $i$  selecting the option  $j$ .

Step 2: Apply Multi-Criteria Decision-Making (MCDM) Locally

For each agent  $i$ :

Use TMF to calculate the individual preferences ( $p_{ij}$ ).

Step 3: Generate the ranking

Generate a priority ranking for each agent by combining the individual preferences ( $p_{ij}$ ) with system-level contributions ( $f_{ij}$ ) into a ranked list.

$$\text{Rank}_{ij} = \alpha \cdot p_{ij} + \beta \cdot f_{ij} \quad (13)$$

Step 4: Aggregate Agent-Level Decisions

After evaluating each agent's options locally, aggregate the top-ranked choices to approximate the system-level solution.

### 2.3.3. Strengths and Limitations

This approach significantly improves scalability by reducing the computational complexity from  $M^N$  (full enumeration) to  $M \cdot N$ , making it feasible for large-scale multi-agent decision-making problems. By treating each agent's decision independently and approximating the system-wide objective as the sum of agent contributions, this method offers a practical alternative to exhaustive search while maintaining reasonable solution quality. Additionally, it allows for parallel decision-making, meaning that each agent can compute its optimal choice separately which guarantees the fairness, leading to faster decision-making in distributed systems.

However, since agents make decisions independently, it does not guarantee a globally optimal solution, as inter-agent dependencies and constraints are not directly accounted for. This can lead to suboptimal system-wide performance, especially in cases where resource constraints or dependencies between agent choices significantly impact the overall objective. Additionally, the approximation of the system-level objective as a sum of individual agent contributions may not always be valid — some global objectives are inherently non-decomposable, meaning that the interactions between agents play a critical role in decision-making.

### 2.4. Approach 3: Iterative Refinement

The Iterative Refinement approach is a hybrid optimization method that balances computational efficiency and solution quality by iteratively improving agent decisions. Unlike the Full Enumeration approach, which explores all possible combinations, and the Decomposition approach, which assumes independence among agents, Iterative Refinement incorporates feedback between the system-level evaluation and agent-level decisions. This feedback loop progressively adjusts decisions to achieve a better balance between individual preferences and system-wide objectives.

#### 2.4.1. Mathematical Model

##### 2.4.1.1. Objectives

(a) Initial Agent-Level Objective

Each agent computes its initial score for all options  $j$  based on a weighted combination of individual preference and system-level contribution like in the Decomposition into Agent Contributions approach:

$$\text{Rank}_{ij} = \alpha \cdot p_{ij} + \beta \cdot f_{ij} \quad (13)$$

$$Z_i = \sum_{j=1}^M \text{Rank}_{ij} \cdot x_{ij} \quad (14)$$

(b) System-Level Objective

Approximate the system-level objective as the sum of agent-level contributions:

$$Z_{\text{system}} = \sum_{i=1}^N Z_i \quad (15)$$

### c) Adjusted Agent-Level Objective

Based on feedback from the system-level evaluation, agents adjust their scores by incorporating penalties for violating system constraints:

$$\text{AdjustedRank}_{ij} = \text{PreviousRank}_{ij} - \zeta \cdot \text{Impact}_{ij} \quad (16)$$

$$Z_i = \sum_{j=1}^M \text{AdjustedRank}_{ij} \cdot x_{ij} \quad (17)$$

where:

$\text{Impact}_{ij}$  : Measures how much agent  $i$ 's decision contributes to constraint violations.

$\zeta$  : Adjusting rate for penalizing decisions with higher system impact.

Key Benefits of saving the previous rank from the previous iteration:

- Smooth Transition Between Iterations: By retaining the  $Z_i$  values from the previous iteration, we ensure that the refinement process builds on the progress made earlier, leading to smoother and more consistent adjustments.

- Better Constraint Handling: The penalties applied in the previous iteration persist in influencing the next iteration, effectively discouraging agents from reverting to previously penalized options.

- Faster Convergence: This memory mechanism reduces oscillations in decision-making, potentially speeding up the convergence to a feasible and optimal solution.

## 2. Constraints

### (a) Individual Agent Constraint

Each agent selects one and only one option:

$$\sum_{j=1}^M x_{ij} = 1, \quad \forall i \in 1, 2, \dots, N \quad (18)$$

### (b) System-Wide Constraints

System resources must remain within their limits:

$$\sum_{i=1}^N \sum_{j=1}^M g_{ijk} \cdot x_{ij} \leq G_k, \quad \forall k \in 1, \dots, K \quad (19)$$

## 3. Solution

The process iteratively refines decisions until a stable, constraint-satisfying solution  $s^*$  is reached, which balances individual and system-wide objectives. We add a final step to recheck the previous decision variables by replacing them one-by-one in the final solution to check if there is a better value of  $Z_{\text{system}}$ .

### 2.4.2. Step-by-Step Workflow

#### Step 1: Initial Solution

Compute Initial Ranks: Each agent evaluates its options using Equation (2.13).

Make Initial Selection: Each agent selects the option  $j^*$  with the highest rank:

$$x_{ij}^0 = \begin{cases} 1, & \text{if } j = \underset{j}{\operatorname{argmax}} \text{Rank}_{ij}, \\ 0, & \text{otherwise.} \end{cases} \quad (20)$$

Form Initial Solution: The initial solution  $s^0$  is the set of decisions  $x_{ij}^0$  for all agents.

#### Step 2: System-Level Evaluation

Check Constraints: Evaluate total resource usage for each constraint  $k$ :

$$\text{Usage}_k = \sum_{i=1}^N \sum_{j=1}^M g_{ijk} \cdot x_{ij} \quad (21)$$

Identify Violations: Compute penalties for constraint violations:

$$\text{Penalty}_k = \max(0, \text{Usage}_k - G_k) \quad (22)$$

#### Step 3: Feedback and Adjustment

Compute Agent Impact: For each agent, compute its contribution to resource usage:

$$\text{Impact}_{ij} = \sum_{k=1}^K \frac{g_{ijk}}{\text{Usage}_k} \cdot \text{Penalty}_k \cdot x_{ij} \quad (23)$$

Adjust Rankings: Penalize options that contribute heavily to constraint violations. The applied penalty will be saved during the iteration:

$$\text{AdjustedRank}_{ij} = \text{PreviousRank}_{ij} - \zeta \cdot \text{Impact}_{ij} \quad (24)$$

Re-select Options: Each agent recomputes its rankings using the adjusted scores and selects the best option:

$$x_{ij}^{\text{new}} = \begin{cases} 1, & \text{if } j = \underset{j}{\operatorname{argmax}} \text{AdjustedRank}_{ij}, \\ 0, & \text{otherwise.} \end{cases} \quad (25)$$

Step 4: Repeat Until Convergence

Convergence Check: Stop the iteration when:

- The system objective  $Z_{system}$  stabilizes:  $|Z_{system}^{new} - Z_{system}^{old}| < \epsilon$ , where  $\epsilon$  is a small tolerance value.
- No agent changes its decision between many iterations:  $\sum_{i=1}^N \sum_{j=1}^M |x_{ij}^{new} - x_{ij}^{old}| = 0$
- Reach the iteration limit.

Output Final Solution:

The final solution  $S^*$  is the set of decisions  $x_{ij}^*$  for all agents.

### 2.4.3. Strengths and Limitations

The Iterative Refinement approach strikes a balance between computational efficiency and solution quality by iteratively improving agent decisions through a feedback mechanism. By continuously refining choices based on violations of system constraints, it achieves a more adaptive and responsive optimization process. A key advantage of this method is that it reduces computational complexity significantly compared to full enumeration while still capturing partial interdependencies between agents, making it more scalable for larger systems. Additionally, the iterative structure helps mitigate constraint violations over time, leading to a feasible and balanced solution without requiring an exhaustive search of the solution space. This process also allows for parallel updates in decentralized settings, making it applicable to multi-agent systems with distributed decision-making.

However, without proper tuning of penalty weights and feedback mechanisms, the iterative process can oscillate between solutions, causing instability. Another limitation is that this approach assumes that all agents respond rationally and iteratively adjust their decisions based on system feedback, which may not hold in dynamic or adversarial environments where agents have strategic behaviors or incomplete information.

### 2.5. Approach 4: Iterative Refinement with Inter-agent Interactions

While the previous approaches addressed many fundamental challenges—such as exponential complexity, balancing local vs. global objectives, and adaptive penalty mechanisms to enforce resource constraints—they did not fully capture how agents' choices interact with each other beyond shared resources—whether they be cooperative synergies, competitive conflicts, or more nuanced relational dynamics. Building on the iterative penalty-based framework of Approach 3, approach 4 closes this gap by incorporating a joint contribution function that quantifies synergy or conflict based on the specific combination of options chosen across agents. In doing so, each agent's objective is no longer limited to local preference, system-level contribution, and resource penalties; it also accounts for how its choice aligns (or clashes) with the decisions of other agents. As a result, the proposed method supports a wide range of cooperative, competitive, and hybrid scenarios, providing a more realistic and dynamic decision-making framework for multi-agent systems.

#### 2.5.1. Mathematical Model

##### 2.5.1.1. Joint Contribution Function for Inter-agent Interactions

###### 2.5.1.1.1. Motivation

When multiple agents coordinate (or compete), the benefit or cost of an agent  $i$  choosing option  $j$  often depends on what other agents  $q \neq i$  select. We capture this via a function  $J_{iq}(j, \ell)$ , which measures the joint effect of agent  $i$  choosing  $j$  while agent  $q$  chooses  $\ell$ .

For example:

Cooperative scenario:  $J_{iq}(j, \ell)$  is positive if  $j = \ell$ , indicating synergy when both agents pick the same option (e.g., same location for easy collaboration).

Competitive scenario:  $J_{iq}(j, \ell)$  might be negative if  $j = \ell$ , penalizing collisions or conflicts (e.g., both agents vying for the same resource).

Partial synergy:  $J_{iq}(j, \ell)$  can take continuous values, reflecting how similar or complementary options  $j$  and  $\ell$  are.

###### 2.5.1.1.2. Interaction Matrix

In addition to  $J_{iq}(j, \ell)$ , let  $I[i, q]$  represent the strength of the relationship between agents  $i$  and  $q$ . A positive  $I[i, q]$  suggests a strongly cooperative link, while a negative value implies strong competition. If agents  $i$  and  $q$  are unrelated,  $I[i, q] \approx 0$ .

Hence the total interaction term for agent  $i$  picking option  $j$  at iteration  $t$  can be expressed:

$$Interaction_{ij}^{(t)} = \sum_{\substack{q=1 \\ q \neq i}}^N I[i, q] \sum_{\ell=1}^M J_{iq}(j, \ell) \cdot x_{q, \ell}^{(t)} \quad (26)$$

###### 2.5.1.1.3. Examples of defining the joint contribution function

We define the Interaction matrix as:

$$I[i, q] = \begin{cases} +1, & \text{if cooperative,} \\ -1, & \text{if competitive,} \\ 0, & \text{if neutral.} \end{cases} \quad (27)$$

Simple joint function  $J_{iq}(j, \ell)$  :

A common binary function:

$$J_{iq}(j, \ell) = \begin{cases} +1, & \text{if } j = \ell, \\ 0, & \text{if } j \neq \ell. \end{cases} \quad (28)$$

Positive  $I[i, q]$  boosts the synergy if both agents match on the same option.

Negative  $I[i, q]$  penalizes matching choices.

Complex joint function  $J_{iq}(j, \ell)$  :

below is a partially rewards (or penalizes) agents  $i$  and  $q$  based on how “compatible” their chosen options  $j$  and  $\ell$  are. The core idea is that each option can be mapped to certain attributes—for instance, a location in physical space, a skill set, or a type label. The synergy or conflict then depends on how similar or complementary these attributes are, not just whether the agents pick the same option.

Example: Location-Based Partial Synergy

Suppose each agent’s options correspond to possible locations (or positions in a metric space). Let:

$\text{pos}_{i,j} \in \mathbb{R}^d$  be the coordinate of agent  $i$  if it chooses option  $j$ .

$d(\text{pos}_{i,j}, \text{pos}_{q,\ell})$  be the distance between these two agents coordinates.

We can design a piecewise function that:

- Rewards pairs of locations that are very close (e.g., good for collaboration).
- Gives a smaller bonus (or neutral effect) if they are moderately distant.
- Penalizes them if they are too far apart (e.g., high communication or transportation cost).

A sample function might look like:

$$J_{ik}(j, \ell) = \begin{cases} +S_{iq} \left[ 1 - \frac{d(\text{pos}_{i,j}, \text{pos}_{q,\ell})}{D_{\max}} \right], & \text{if } d(\text{pos}_{i,j}, \text{pos}_{q,\ell}) \leq \delta, \\ 0, & \delta < d(\text{pos}_{i,j}, \text{pos}_{q,\ell}) \leq \gamma, \\ -C_{ik}, & \text{if } d(\text{pos}_{i,j}, \text{pos}_{q,\ell}) > \gamma. \end{cases} \quad (29)$$

Where:

$S_{iq}$  : Synergy factor for agents  $i$  and  $q$ . If they are close enough to collaborate effectively, they gain up to  $S_{iq}$  points—scaled by how close they are relative to the maximum expected distance  $D_{\max}$ .

$\delta$  is a proximity threshold below which agents gain a partial synergy bonus. The closer they are, the larger the reward (linearly scaled).

$\gamma$  is a penalty threshold above which agents are considered “too distant” for collaboration.

$C_{iq}$  : Conflict cost.

For intermediate distances  $\delta < d(\cdot) \leq \gamma$ , the effect is neutral (zero).

Other Possible Joint contribution function formulations

- Attribute Similarity: Instead of location distance, we could define a function based on the similarity of agents’ skill sets, technology types, or preferences. For instance, if  $\text{sim}(j, \ell)$  is a cosine similarity between two option vectors, let

$J_{iq}(j, \ell) = A_{iq} \times \text{sim}(j, \ell) - B_{iq}$ , where  $A_{iq}$  is a synergy scaling and  $B_{iq}$  is a baseline conflict level.

- Nonlinear or Multi-Attribute: We might combine multiple factors—distance + type alignment + resource overlap—into a single function, weighting each sub-factor differently.

- Time-Dependent: For dynamic environments, let  $J_{iq}(j, \ell; t)$  evolve over time if agent  $i$  and  $q$  become more (or less) willing to coordinate after repeated interactions.

- Probabilistic: Define expected synergy or conflict, e.g.  $J_{iq}(j, \ell) = \mathbb{E}[\dots | j, \ell]$ , where random events (e.g., uncertain costs or random external factors) influence the payoff.

### 2.5.1.2. Refined Objective Function

Agent-Level Objective

At iteration  $t$ , each agent  $i$  forms an objective that combines:

Local preference ( $p_{ij}$ ),

System-level contribution ( $f_{ij}$ ),

Penalty for resource overuse (adaptive from iteration  $t$ ),

Interaction with other agents.

The general objective for agent  $i$  at iteration  $t$  :

$$\begin{aligned} Z_i^{(t)}(x_{ij}) = & w_1 \sum_{j=1}^M p_{ij} \cdot x_{ij} + w_2 \sum_{j=1}^M f_{ij} \cdot x_{ij} \\ & - w_3 \sum_{j=1}^M \text{Impact}_{ij}^{(t)} \cdot x_{ij} + w_4 \sum_{j=1}^M \text{Interaction}_{ij}^{(t)} \cdot x_{ij} \end{aligned} \quad (30)$$

- $w_1, w_2$  : Weights for preference and system-level terms. (Same as  $\alpha, \beta$  in the previous approaches).  
 $w_3$  : Penalty weight. (Same as  $\zeta$  in the third approach).  
 $w_4$  : Weight for the interaction term (positive if synergy is desired, negative if conflict should be strongly penalized).  
 $Impact_{ij}^{(t)}$  quantifies how selecting option  $j$  for agent  $i$  contributes to resource violations.  
 $Interaction_{ij}^{(t)}$  quantifies the cooperative or competitive relationships with other agents.

Each agent  $i$  then chooses the option  $j$  that maximizes this local objective.

### 2.5.2. Step-by-step workflow: Iterative Refinement Process

The algorithm proceeds in iterations  $t = 0, 1, 2, \dots$  until convergence or a maximum iteration limit.

#### Step 1: Initialization

- Set an initial decision matrix  $X^{(0)} = x_{ij}^{(0)}$  where each agent picks the option that maximizes  $w_1 \cdot p_{ij} + w_2 \cdot f_{ij}$  ignoring constraints or interactions (a “greedy” start).
- Initialize  $Impact_{ij}^{(0)} = 0$  or some neutral baseline.

$t \leftarrow 0$ .

#### Step 2: Compute Resource Usage & Penalties

- For each resource  $k$ , compute  $Usage_k^{(t)}$ .
- Determine  $Penalty_k^{(t)}$ .
- Allocate impacts based on  $Penalty_k^{(t)}$  to each option selected by each agent.

#### Step 3: Compute Interaction Terms

For each agent  $i$ , with a selected option  $j$ : calculate the interaction value using Equation (2.26).

#### Step 4: Agent-Level Optimization

Each agent  $i$ :

$$x_{ij}^{(t+1)} = \begin{cases} 1, & \text{if } j = \arg \max_j Z_i^{(t)}(x_{ij}) \\ 0, & \text{otherwise.} \end{cases} \quad (31)$$

where  $Z_i^{(t)}$  uses the updated impacts and interactions from Steps 2–3.

#### Step 5: Convergence Check

Check if:

- The total system-level objective (or sum of all  $Z_i^{(t)}$ ) does not significantly change, e.g.,  $|Z_{system}^{(t+1)} - Z_{system}^{(t)}| < \epsilon$ .
- All resource constraints are satisfied (or the penalty remains unchanged).
- No agent changes its decision, i.e.,  $\sum_{i,j} |x_{ij}^{t+1} - x_{ij}^t| = 0$ .

If these conditions are met, stop. Otherwise, increment  $t$  and repeat from Step 2.

### 2.5.3. Strengths and Limitations

Approach 4’s main advantage is its ability to capture a wide range of cooperative or competitive inter-agent dynamics by introducing explicit interaction terms beyond simple resource-sharing constraints. It adaptively refines decisions in an iterative manner, allowing for parallel agent updates and offering greater scalability than exhaustive methods, while still preserving a multi-criteria perspective by combining individual preferences, system-level contributions, penalty terms, and interaction factors into a unified objective. However, tuning the additional interaction weights and penalty parameters introduces further complexity and can lead to oscillations if not carefully calibrated. The accuracy of the results relies heavily on defining the joint function. Finally, in highly adversarial or rapidly changing environments, agents may not faithfully follow the iterative penalty updates, and more robust game-theoretic or reinforcement learning approaches may be required for stable and fair outcomes.

### Verification using the Rastrigin Function

The Rastrigin function was selected as a representative benchmark for evaluating the performance of the proposed MAS-MCDM approaches. This function is traditionally used in single-agent global optimization to test an algorithm’s ability to escape local minima. It was used to verify the proposed models for a multi-agent scenario by discretizing it. These discretizing points represent the agent’s options.

#### 3.1. Implementation Overview

##### 3.1.1. Discretized Options

For each agent  $i \in 1, \dots, N$ , where  $N = 5$ , a set of equally spaced sample points were generated between  $[-5.12, 5.12]$ . Each generated set represents a set of options for one agent  $\{x_{i1}, \dots, x_{iM}\}$ . Generally, the global minimum occurs at  $x = 0$ . Due to the constraints or interactions between agents, the final solution can be shifted.

The generated points (Set of agent’s options) in this implementation are:  $[-5.12, -4.096, -3.072, -2.048, -1.024, 0, 1.024, 2.048, 3.072, 4.096, 5.12]$

##### 3.1.2. Preference, Contribution, and Resources

For each agent  $i$  and option  $j$ , the Rastrigin term is calculated as in Equation:



$$\text{RastriginTerm}(x) = x^2 - A \cdot \cos(2\pi x), \quad \text{where } A = 10. \quad (32)$$

Then assign:  $p_{ij} = f_{ij} = g_{ij} = -\text{RastriginTerm}(x_j)$  to convert a minimization problem into a maximization objective (  $p_{ij}$  and  $f_{ij}$  become larger when  $x_j$  is closer to the global optimum near zero). Figure 1 shows a plot of the Rastrigin function with  $A = 10$ .

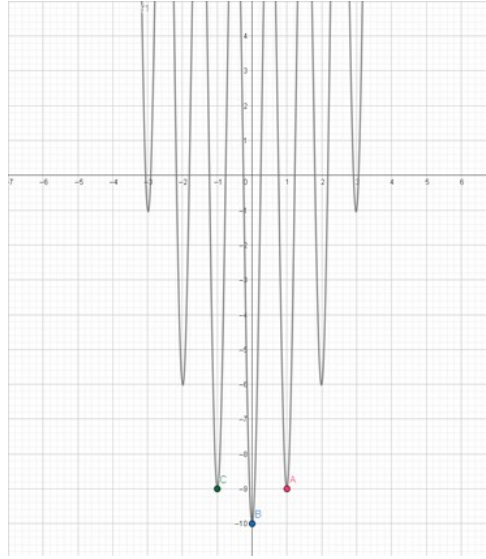


Figure 1 - The Rastrigin function with parameter  $A=10$ . Point B at  $x=0$  is the continuous global minimum in single-agent scenarios; in multi-agent settings, additional constraints and interactions can shift the solution away from  $x=0$  to A and C points

DOI: <https://doi.org/10.60797/IRJ.2025.158.94.1>

### 3.1.3. Objectives Weighting

The weights  $\alpha = \beta = w_1 = w_2 = 0.5$  and additional penalty/inter-agent weights (  $\zeta = w_3, w_4$  ) vary depending on how strongly we want constraints or synergy to influence decisions. This balances each agent's local preference with the system-level contribution equally. In practice, these weights can be adjusted to reflect different priorities between individual agents and the overall system.

### 3.1.4. Constraints

- **Resource Constraints:** Resource-usage terms derived from the negative Rastrigin values and impose a global limit  $G$ . This can push agents away from the global minimum at  $x = 0$  when many agents choose it simultaneously.

- Let the global resource threshold  $G = 45$ , the usage of resources for each agent at the global minimum is 10, and for 5 agents the total usage is 50.

- **Interaction Constraints:** Reward-based interactions were incorporated in the implementation of Approach 4, which can steer agents to local minima near  $\pm 1$  or shape cooperative/competitive behaviors.

### 3.1.5. Inter-Agent Interactions:

#### 3.1.5.1. Interaction Setup

We created a scenario where first and fifth agents set as the first group and the other agents set as the second group, each group benefit from matching their values sign (cooperative between agents in the same group). Additionally, cross-group interactions are set as conflict, discouraging all five from picking the same sign (competitive between groups).

Interaction matrix:

$$I = \begin{bmatrix} 0 & -1 & -1 & -1 & 1 \\ -1 & 0 & 1 & 1 & -1 \\ -1 & 1 & 0 & 1 & -1 \\ -1 & 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & -1 & 0 \end{bmatrix}$$

#### 3.1.5.1. Normalization

The normalized preferences and contributions onto  $[0, 1]$ , and interaction terms onto  $[-1, 1]$ , to prevent the large penalty or interaction magnitudes from overshadowing the individual preference. This normalization step is executed at problem formulating or after computing matrix terms, is crucial to obtaining stable and interpretable solutions.

### 3.2. Results and Observations

#### - Approach 1 (Full Enumeration)

Outcome:

Correctly identifies the global minimum  $x = 0$  for all agents when unconstrained. And the objective value  $Z = 50$ .

With Constraints:

When resource usage rules push agents away from  $x = 0$ , the solver's exhaustive search locates the best feasible combination (e.g., local minima near  $\pm 1$ ). In our implementation, we found 39 combinations that satisfy the constraints with total usage value equals 44.19.

Runtime: Exponential in  $M^N$ , feasible only for small  $N$  and  $M$ .

Despite guaranteeing the global optimum, the exponential complexity restricts this method to small values of  $N$  and  $M$ , making it a good choice for comparison.

- *Approach 2 (Decomposition)*

Outcome:

Each agent individually picks the best discrete point, ignoring synergy or resource constraints. All agents choose  $x = 0$ .

- *Approach 3 (Iterative Refinement)*

Outcome:

Agents iteratively adjust choices based on feedback from resource penalties until finding the solution where satisfy the constraints.

Resource  $G = 45$ : All agents must shift to  $\pm 1$ . In our results, all agents adjusted their choices based on their impacts, which lead from the first iteration to find the local optimal solution at  $x = -1.024$  for all agents.

This demonstrates that iterative penalty-based methods handle resource constraints effectively without enumerating all solutions.

- *Approach 4 (Iterative Refinement with Inter-Agent Interactions)*

Outcome:

There are two combinations that solve the problem of constraints and interactions. This approach found a combination that satisfy the constraints with total usage value equals 44.19. The both agents in the first group selected  $+1.024$ , and all agents in the second group selected  $-1.024$ .

As shown, incorporating inter-agent synergy and conflict can yield solutions where agent groups naturally self-organize, surpassing purely penalty-based approaches in scenarios with strong interactions.

The Rastrigin experiments validate that each proposed approach can be adapted to MAS-MCDM, handling both simpler unconstrained cases and more intricate interactions/constraints. The results underscore the trade-offs between optimality, scalability, and the capacity to model inter-agent effects — key considerations for multi-agent systems in many practical domains.

## Conclusion

In this paper, we have presented a comprehensive framework for MAS MCDM to effectively balance local agent preferences with system-wide objectives and constraints. We introduced and rigorously formulated four distinct approaches — full enumeration, decomposition into agent contributions, iterative refinement, and iterative refinement with inter-agent interactions — each offering a unique set of trade-offs between global optimality and computational scalability. This work systematically demonstrates that while exhaustive search guarantees a global optimum, its exponential complexity renders it impractical for large-scale systems, thereby motivating the need for more scalable methods.

A key contribution of the research is the novel iterative penalty-based refinement strategy. This mechanism dynamically adjusts agent-level decisions based on feedback from resource constraint violations and, in the most advanced approach, inter-agent interactions. By integrating these dynamic adjustments, the framework is capable of converging to robust and feasible solutions even in the presence of complex interdependencies and resource limitations. The incorporation of explicit interaction terms further allows the model to capture cooperative and competitive dynamics, thereby enhancing its applicability to a wide range of real-world scenarios — from tightly coupled, mission-critical systems to decentralized and distributed decision-making. The validation experiments using a discretized version of the Rastrigin function underscore the efficacy of the proposed methods. The experimental results highlight the strengths of each approach and illuminate the inherent trade-offs between achieving optimality and ensuring scalability. In particular, the iterative refinement strategies have shown promise in navigating challenging optimization landscapes. Future work may investigate further enhancements to the interaction models, integrate with reinforcement learning and game-theoretic strategies, and apply these methods to real-world challenges in autonomous systems, resource management, and other areas.

## Конфликт интересов

Не указан.

## Рецензия

Борисов А.Н., Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ, Казань Российская Федерация  
DOI: <https://doi.org/10.60797/IRJ.2025.158.94.2>

## Conflict of Interest

None declared.

## Review

Borisov A.N., Kazan National Research Technical University named after A.N. Tupolev – KAI, Kazan Russian Federation  
DOI: <https://doi.org/10.60797/IRJ.2025.158.94.2>

## Список литературы на английском языке / References in English

1. Azhar N.A. Multi-criteria Decision Making: A Systematic Review. / N.A. Azhar, N.A. Mohamed Radzi, W.S.H.M. Wan Ahmad // Recent Advances in Electrical & Electronic Engineering. — 2021. — № 14. — DOI: 10.2174/2352096514666211029112443

2. Basílio M.P. A Systematic Review of the Applications of Multi-Criteria Decision Aid Methods (1977–2022). / M.P. Basílio, V. Pereira, H.G. Costa et al. // *Electronics*. — 2022. — № 11. — P. 1720. — DOI: 10.3390/electronics11111720
3. Xing Y. A bargaining game based feedback mechanism to support consensus in dynamic social network group decision making. / Y. Xing, J. Wu, F. Chiclana et al. // *Information Fusion*. — 2023. — № 93. — P. 363–382. — DOI: 10.1016/j.inffus.2023.01.004
4. Li G. Heterogeneous Large-Scale Group Decision Making Using Fuzzy Cluster Analysis and Its Application to Emergency Response Plan Selection. / G. Li, G. Kou, Y. Peng // *IEEE Transactions on Systems, Man, and Cybernetics: Systems*. — 2022. — № 52. — P. 3391–3403. — DOI: 10.1109/TSMC.2021.3068759
5. Dorri A. Multi-Agent Systems: A Survey. / A. Dorri, S.S. Kanhere, R. Jurdak // *IEEE Access*. — 2018. — № 6. — P. 28573–28593. — DOI: 10.1109/ACCESS.2018.2831228
6. Russell S. Artificial Intelligence: A Modern Approach / S. Russell, P. Norvig. — Pearson, 2020. — 1136 p.
7. Jin W. A Comprehensive Survey on Multi-Agent Cooperative Decision-Making: Scenarios, Approaches, Challenges and Perspectives / W. Jin, H. Du, B. Zhao et al. // *arXiv*. — 2025. — URL: <https://arxiv.org/abs/2503.13415> (accessed: 03.04.25). — DOI: 10.48550/arXiv.2503.13415.
8. Macharis C. Reviewing the use of Multi-Criteria Decision Analysis for the evaluation of transport projects: Time for a multi-actor approach. / C. Macharis, A. Bernardini // *Transport Policy*. — 2015. — № 37. — P. 177–186. — DOI: 10.1016/j.tranpol.2014.11.002
9. Hammoud A. Trimetric Fusion: A Novel Algorithm For Multi-Criteria Decision-Making. / A. Hammoud // *International Research Journal*. — 2025. — № 3. — DOI: 10.60797/IRJ.2025.153.6
10. Chaube Sh. An Overview of Multi-Criteria Decision Analysis and the Applications of AHP and TOPSIS Methods / Sh. Chaube, A. Kumar, M. Singh [et al.] // *International Journal of Mathematical Engineering and Management Sciences*. — 2024. — № 9. — P. 581–615. — DOI: 10.33889/IJMEMS.2024.9.3.030.
11. Korobko A.S. Comparison of Evolutionary Optimization Methods on the Example of Rastrigin Function / A.S. Korobko, T.T. Gazizov, M. Moskalev // *2024 IEEE Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT)*. — 2024. — P. 49–53. — DOI: 10.1109/USBEREIT61901.2024.10584012