Research Article *Open Access*

Journal of Pioneering Artificial Intelligence Research

ISSN: 3069-0846

DOI: *doi.org/10.63721/26JPAIR0136*

Deterministic Goal Selection via Intrinsic and Contextual Constraint-Governed Elimination

Huseyin Murat Cekirge

Department of Mechanical Engineering, The City College of New York (CUNY), New York, USA

Citation: *Huseyin Murat Cekirge (2026) Deterministic Goal Selection via Intrinsic and Contextual Constraint-Governed Elimination. J of Poin Artf Research 2(3), 1-14*
WMJ-JPAIR-136

Abstract

Autonomous decision-making systems commonly assume that the objective to be pursued is predefined, focusing computational effort on generating action sequences or policies that satisfy a given goal. This work introduces a deterministic framework in which goal selection itself is treated as a primary computational process rather than an external input. The proposed approach operates over a structured internal goal library, referred to as the identity layer, which defines a bounded set of admissible objectives together with persistent constraints that remain active across all operating conditions. At each time step, the system evaluates its current context, composed of environmental conditions, internal state variables, and operational role, and maps this context to a set of constraints that restrict the admissibility of goals. Instead of evaluating or ranking all candidate goals, the framework eliminates those that violate either identity-level constraints or context-derived constraints, producing a reduced set of admissible goals. If a single goal remains, it is selected directly; if multiple admissible goals persist, a deterministic residual resolution mechanism is applied using simple ordering or priority rules within the reduced set. In cases where no admissible goal remains, predefined safe-state objectives are activated. This formulation shifts decision-making from optimization and stochastic exploration to constraint-driven reduction, ensuring that all selected goals are consistent with both internal structure and external conditions. The approach does not rely on reward functions, probabilistic policies, or global search, and therefore maintains low computational complexity and high interpretability. It is designed to operate as a pre-planning layer for existing systems such as Goal-Oriented Action Planning, providing a deterministic mechanism for generating context-consistent goals prior to action planning. Compared with reinforcement learning and optimization-based methods, the proposed framework emphasizes admissibility over optimality, making it particularly suitable for safety-critical, real-time, and resource-constrained applications. By structuring goal selection as a process of elimination under constraints, the framework provides a unified and extensible basis for autonomous operation in environments where consistency, predictability, and bounded behavior are essential. Agents operate with evolving behavioral libraries that are updated through experience while remaining constrained by objective and governing principles. This enables swarm-like coordination under minimal supervision, where the selection emerges through constraint-consistent elimination over a dynamically maintained action space. A self-check mechanism enforces internal consistency by evaluating admissible actions against recent behavior and operational intent, ensuring coherent and stable autonomous operation.

*Corresponding author: Huseyin Murat Cekirge, Department of Mechanical Engineering, The City College of New York (CUNY), New York, USA. E-mail: hmcekirge@usa.net

Submitted: 30.04.2026 **Accepted**: 04.05.2026 **Published**: 15.05.2026

Keywords: Deterministic Goal Selection, Context-Driven Constraints, Identity Layer, Admissibility Filtering, Goal Space Elimination, Autonomous Systems, GOAP, Real-Time Decision-Making, Safety-Critical Systems, Constraint-Based Reasoning

Abbreviations

AI	Artificial Intelligence
GOAP	Goal-Oriented Action Planning
RL	Reinforcement Learning
G	Goal library (set of candidate goals)
Π	Persistent constraint set (identity layer)
K	Context
C(K)	Context-derived constraint set
G_valid	Admissible goal set
G_safe	Safe-state goal subset
Π ∧ C(K)	Combined constraint set (identity + context)

Introduction

Autonomous decision-making systems typically assume that the objective to be achieved is predefined. In widely used frameworks such as Goal-Oriented Action Planning (GOAP), the system is provided with a goal state, and the primary task is to determine an action sequence that satisfies this objective under given constraints. While effective in structured environments, this assumption shifts the most critical question of which goal should be pursued outside the computational framework.

In parallel, approaches based on Reinforcement Learning (RL) address decision-making through reward maximization, where policies are learned via interaction with the environment. Although powerful, such methods rely on reward function design, stochastic exploration, and iterative convergence, which may introduce variability and reduce interpretability, particularly in safety-critical or real-time applications. Similarly, classical optimization-based methods formulate decision-making as a global search problem over a cost function, requiring evaluation across potentially large solution spaces and careful tuning of objective criteria [1-4].

These paradigms share a common structural characteristic: **they operate on a predefined or externally specified objective space**. The process of determining admissible goals is either assumed, learned indirectly, or embedded in heuristic design. As a result, the goal selection stage remains weakly formalized.

This work introduces a deterministic framework in which goal selection is treated as a primary computational process rather than a predefined input. The central premise is that goals are not selected through global evaluation or stochastic exploration, but **emerge through constraint-driven elimination**. At any given time, the system maintains a structured library of candidate goals. The current context comprising environmental conditions, internal state, and operational role induces a set of constraints that remove infeasible goals from this library. The remaining admissible set defines the space within which selection occurs.

Formally, the framework consists of three layers:

1. **Identity layer**: a structured internal library defining admissible goals and persistent constraints
2. **Context mapping**: transformation of observed conditions into constraint sets
3. **Deterministic elimination and selection**: reduction of the goal space followed by resolution within the admissible set

This structure shifts decision-making from evaluation to reduction. Instead of scoring all possible objectives, the system eliminates those that violate constraints, resulting in a bounded and consistent set of admissible goals. Selection, if required, is performed only within this reduced set and does not require stochastic or global optimization.

The proposed formulation complements existing planning systems. In particular, it operates as a pre-planning layer for GOAP-like frameworks, providing a consistent and deterministic mechanism for generating goal states prior to action planning. This separation improves interpretability and reduces ambiguity in downstream decision processes.

The motivation for this approach arises from domains where constraints dominate behavior and must be enforced deterministically, including autonomous robotics, safety-critical control systems, and resource-limited embedded agents [5-7]. In such systems, admissibility is often more critical than optimality, and the ability to exclude infeasible or unsafe goals prior to planning can significantly improve robustness.

The contribution of this work is therefore not a new optimization method or learning algorithm, but a **structural reformulation of goal selection**. By treating context as a generator of constraints and identity as a bounded goal space, the framework provides a deterministic and interpretable basis for autonomous decision-making that is compatible with existing planning architectures.

In practical systems, agents do not rely on continuous command but operate using evolving behavioral libraries constructed from prior interactions and observations. These libraries support swarm-like coordination under minimal supervision, where collective behavior emerges from local interactions. Despite their adaptive nature, such libraries remain constrained by the agent's objective and governing principles, ensuring coherent and bounded decision-making.

Decision-making in autonomous systems is often interpreted as the selection of goals from a set of available alternatives. However, in structured systems, this selection is not unconstrained. From the outset, admissible behavior is bounded by intrinsic constraints that define the system's identity, while contextual conditions introduce additional, time-varying restrictions. As a result, the space of possible goals is not freely explored but progressively reduced through constraint-consistent elimination. In this sense, autonomy does not imply unrestricted choice, but rather the emergence of behavior within a constrained and well-defined admissible space.

Creation of the Identity Layer and Library Elements

Definition of the Identity Layer

The proposed framework introduces an internal identity layer that defines the admissible behavioral space of the agent. This layer is not an external supervisory mechanism, but an embedded structural component that constrains decision-making.

The identity layer is represented as a structured internal library consisting of:

- A predefined set of candidate goals
- A set of persistent constraints governing admissibility

This structure eliminates the need for centralized control or external supervision. Instead, all decisions are bounded by the internal consistency of the identity layer.

Formally, the identity layer is defined as:

- Goal library:
  $G = \{g_1, g_2, \ldots, g_n\}$
- Persistent constraint set:
  $\Pi = \{\pi_1, \pi_2, \ldots, \pi_k\}$

The identity of the agent is therefore not a descriptive attribute, but a functional structure that determines which goals are admissible under all conditions.

Role of the Identity Layer in Decision Formation

The identity layer operates as a persistent admissibility filter. It does not select goals directly, nor does it evaluate them through optimization. Instead, it removes goals that violate invariant constraints.

Given a context-dependent constraint set C(K), derived from the current state and environment, the admissible goal set is defined as:

$$G\_valid = \{ g \in G \text{ such that } g \text{ satisfies } \Pi \text{ and } C(K) \}$$

This formulation ensures that all candidate goals are first filtered through:

1. Internal identity constraints (Π), and
2. Context-dependent constraints (C(K))

Only goals consistent with both sets remain eligible for selection.

Thus, the identity layer defines the boundary of possible behavior, while the context determines the subset of goals that are currently feasible.

Composition of the Identity Library

The identity library is composed of structured elements that collectively define the operational character of the agent. These elements are not arbitrary; they must be explicitly constructed to ensure determinism and consistency.

The library includes:

Goal Set (G)

A finite and structured set of admissible goal states, representing all possible objectives the agent can pursue.

Examples include:

- Task execution goals
- Maintenance or recovery goals
- Safety-related fallback goals

Persistent Constraints (Π)

A set of invariant constraints that are always active, independent of context. These include:

- **Self-preservation constraints**
  (energy limits, structural integrity, operational continuity)
- **Inter-agent compatibility constraints**
  (collision avoidance, system consistency, coordination rules)
- **Mission-level constraints**
  (task boundaries, operational limits, role definitions)

These constraints are not evaluated probabilistically; they act as binary filters.

Construction Principles

The identity library must satisfy the following properties:

- **Boundedness**
  The goal set must be finite or discretized to allow deterministic elimination.
- **Consistency**
  Persistent constraints must not conflict in a way that eliminates all goals under normal operating conditions.
- **Completeness**
  The goal set must include fallback or safe-state goals to handle infeasible scenarios.
- **Extensibility**
  The library may be expanded or modified without altering the underlying decision mechanism.

Absence of External Supervision

A key property of the identity layer is the absence of external control. The system does not rely on a centralized authority to enforce behavior.

Instead:

- Admissibility is encoded internally
- Constraints are continuously enforced through elimination
- Decision-making remains deterministic

This eliminates the need for:

- Stochastic exploration
- External rule enforcement
- Centralized monitoring systems

Summary

The identity layer defines the structural foundation of goal selection. It provides:

- A bounded goal space
- A persistent constraint system
- A deterministic admissibility mechanism

As a result, the agent does not select goals freely. It operates within a predefined identity structure, where infeasible goals are removed and only admissible objectives remain available for execution. The identity layer does not prescribe behavior; it defines the limits within which behavior can emerge.

Context to Constraint Mapping

Definition of Context

In the proposed framework, context is defined as the set of observable and internal conditions that influence the admissibility of goals at a given time.

Context is not treated as a descriptive label (e.g., “school”, “vehicle”, “combat”), but as a structured input that generates constraints. It is composed of three components:

- **Environmental context**
  (physical surroundings, objects, spatial conditions)
- **Internal state context**
  (energy level, system status, structural condition)
- **Role context**
  (assigned function, operational mode, mission

state)
Let the context at time t be denoted as:
K(t)

Figure 1 illustrates how context-derived constraints reduce the goal space defined by the identity layer, yielding a single admissible objective that is subsequently provided to the planning module.

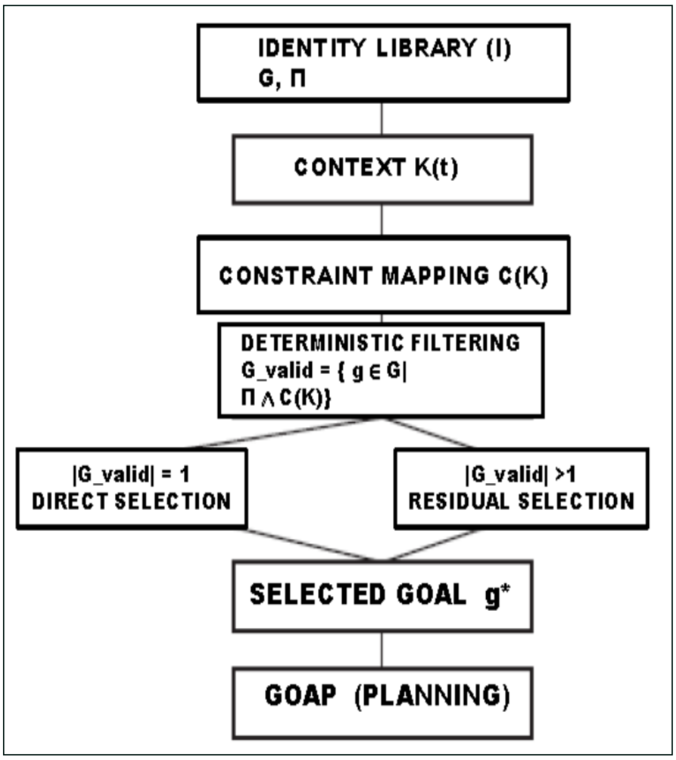

Figure 1: Deterministic goal selection pipeline in which identity and context jointly impose constraints that eliminate infeasible goals, leaving a single admissible objective for downstream planning.

Context-Induced Constraints

The primary function of context is to generate a set of constraints that restrict the admissibility of goals.

Define a context-to-constraint mapping:
$C(K): K \rightarrow \{c_1, c_2, \ldots, c_m\}$

Each constraint c_i acts as a binary filter:
$c_i(g) \subseteq \{0,1\}$

where:

- $c_i(g) = 1$ indicates that goal g is admissible under constraint c_i
- $c_i(g) = 0$ indicates violation and immediate elimination

Thus, context does not prescribe behavior directly; it defines the constraint structure under which goals are filtered.

Interaction with the Identity Layer

The admissibility of goals is determined by the joint action of:

- Persistent constraints Π (identity layer)
- Context-derived constraints C(K)

The resulting valid goal set is:
G_valid = { g ∈ G such that g satisfies Π and C(K) }

This ensures that:

- Identity defines what is always permissible
- Context defines what is currently feasible

Only goals consistent with both are retained.

Constraint Categories

Context-derived constraints can be grouped into functional categories:

Physical feasibility constraints

- Motion limits
- Accessibility
- Environmental compatibility

Resource Constraints

- Energy availability
- Time limitations
- Computational capacity

Safety Constraints

- Proximity to hazards
- Violation of safe operating bounds

Temporal Consistency Constraints

- Persistence over time
- Continuity of observation
- Stability of state transitions

Each category contributes to the progressive elimination of infeasible goals.

Deterministic Filtering Process

Given a goal library G, context induces successive filtering:
$G_0 = G$
G_1 = filtered by c_1

G_2 = filtered by c_2
…
G_valid = G*

where:
$G_{k+1} \subseteq G_k$

The process terminates when:
- |G_valid| = 1 → a unique admissible goal remains
- |G_valid| > 1 → residual selection is required
- |G_valid| = 0 → fallback or safe-state activation

This filtering process is deterministic and does not involve:
- Probabilistic inference
- Optimization
- Stochastic exploration

Example of Context Mapping

Consider an agent with candidate goals:
G = {continue task, explore, recharge, retreat}

Case 1: Low-Energy Condition
Context:
Internal state: energy below threshold

Constraint:
Eliminate goals requiring high energy

Result:
G_valid = {recharge}

Case 2: High threat condition
Context:
- Environmental: approaching hazard
- Role: survival priority

Constraints:
- Eliminate exposure-increasing goals

Result:
G_valid = {retreat}

Case 3: Nominal operation
Context:
- Stable environment
- Sufficient energy

Constraints:
- minimal filtering

Result:
G_valid = {continue task, explore}
Residual selection may apply.

Absence of Context as Instruction

A key property of the framework is that context does not define goals explicitly.

Instead:
- Context eliminates incompatible goals
- Admissible goals emerge as survivors

Thus, behavior is not instructed, but constrained.

Summary
The context-to-constraint mapping provides the dynamic component of the framework. It:
- Transforms observable conditions into constraint sets
- Interacts with the identity layer to define admissibility
- Enables deterministic reduction of the goal space

As a result, goal selection is not driven by evaluation or optimization, but by the progressive elimination of goals that violate context-induced constraints. Context does not select goals; it removes those that cannot be sustained.

Deterministic Goal Selection and Residual Resolution

From Admissibility to Selection

Following the identity-layer filtering (Section 2) and context-to-constraint mapping (Section 3), the system produces an admissible goal set:
G_valid = { g ∈ G such that g satisfies Π and C(K) }

This set represents all goals that are both structurally permissible and contextually feasible. The role of this section is to define how a final goal is obtained from G_valid without introducing stochasticity or global optimization.

Deterministic Selection Regimes

The selection process operates under three regimes:

Case 1: Unique Admissible Goal
I f :
|G_valid| = 1

Then:
- The remaining goal is selected directly

- No further processing is required

This represents full resolution through constraint elimination.

Case 2: Multiple Admissible Goals

If:
|G_valid| > 1

Then:
- All remaining goals satisfy Π and C(K)
- Ambiguity is limited to admissible alternatives

A deterministic residual selection rule is applied.

Case 3: Empty Admissible Set

If:
|G_valid| = 0

Then:
- All candidate goals violate constraints
- The system enters a fallback condition

Fallback is defined within the identity library (Section 2) as a safe-state goal set G_safe ⊆ G.

Residual Resolution Mechanism

Residual resolution applies only when multiple admissible goals remain. It must satisfy:
- Determinism
- Low computational cost
- Consistency with identity constraints

No stochastic sampling or global optimization is permitted.

Deterministic selection rules may include:

1. **Priority ordering**
   A predefined ordering over G assigns precedence:
   $g_1 \succ g_2 \succ \ldots \succ g_n$
2. **Urgency-based ordering**
   Goals are ranked using simple scalar measures (e.g., time-to-failure, resource depletion)
3. **Lexicographic selection**
   Goals are evaluated sequentially across ordered criteria until differentiation occurs
4. **Minimal-cost local heuristic (optional)**
   Applied only within G_valid, not across the full goal space

These rules operate on a reduced set and therefore remain computationally tractable.

Separation from Optimization-Based Methods

The proposed selection mechanism differs fundamentally from optimization-based approaches:
- No global search over G
- No probabilistic evaluation
- No reward maximization

Instead:
- Infeasible goals are eliminated first
- Selection is applied only to admissible goals

If heuristic evaluation is used, it is confined to G_valid and does not alter the deterministic nature of elimination.

Integration with Planning Systems

Once a goal g* is selected, it is passed to a downstream planning module, such as Goal-Oriented Action Planning (GOAP).

Thus:
Context → Constraints → Goal Elimination → Goal Selection → Planning

The proposed framework operates as a pre-planning layer, reducing the goal space before planning begins.

This separation ensures:
- Reduced planning complexity
- Improved real-time performance
- Consistency between goal and feasible execution

Temporal Consistency and Re-evaluation

Goal selection is not static. At each time step:
- Context K(t) is updated
- Constraints C(K(t)) are recomputed
- G_valid is re-evaluated

If:
$g^* \notin$ G_valid at time t+1

Then:
- The current goal is invalidated
- A new selection process is triggered

This enables adaptive behavior without requiring continuous optimization.

Summary

Deterministic goal selection is achieved through:
1. Elimination of infeasible goals
2. Identification of an admissible goal set
3. Deterministic resolution within that set

The system does not select goals freely. Selection occurs only within the constraint-defined admissible space, ensuring that all chosen goals are consistent with both identity and context. Selection does not create a goal; it resolves among those that remain admissible.

Comparison with GOAP, Reinforcement Learning, and Optimization-Based Methods

Overview

The proposed framework addresses a stage of decision-making that is typically assumed to be given in existing approaches: the definition of the goal itself. Rather than optimizing over actions toward a predefined objective, the framework determines which goals are admissible prior to planning.

This section contrasts the proposed method with three widely used paradigms:
- Goal-Oriented Action Planning (GOAP)
- Reinforcement Learning (RL)
- Optimization-based decision frameworks

Comparison with GOAP

GOAP operates under the assumption that a valid goal state is already defined. Its primary function is to compute an action sequence that transitions the system from the current state to the specified goal.

In contrast, the proposed framework introduces a pre-planning layer that determines the goal itself through deterministic elimination.

GOAP characteristics:
- Input: predefined goal
- Method: search over action sequences
- Objective: reach goal with minimal cost or steps

Proposed framework:
- Input: goal library and context
- Method: constraint-based elimination
- Output: admissible goal for planning

Thus, the two approaches are complementary:
- The proposed method determines **which goal should be pursued**
- GOAP determines **how to achieve that goal**

This separation reduces ambiguity at the planning stage and ensures that the selected goal is consistent with both system constraints and context.

Comparison with Reinforcement Learning

Reinforcement Learning (RL) selects actions through interaction with an environment, guided by a reward function. Behavior emerges from the maximization of expected cumulative reward, typically requiring exploration and iterative updates.

In contrast, the proposed framework:
- Does not use reward functions
- Does not rely on stochastic exploration
- Does not require training or convergence

RL characteristics:
- Decision mechanism: reward maximization
- Behavior: learned through exploration
- Uncertainty: inherent due to stochastic policies

Proposed framework:
- Decision mechanism: constraint satisfaction
- Behavior: determined through elimination
- Uncertainty: minimized through determinism

While RL is effective in environments with unknown dynamics or where learning is required, it introduces variability and requires significant computational resources.

The proposed framework is more suitable for:
- Safety-critical systems
- Real-time applications
- Environments where constraints are well-defined

Comparison with Optimization-Based Methods

Optimization-based methods define decision-making as a global search problem over a cost or objective function. Solutions are obtained by minimizing or maximizing a scalar criterion, often across a large search space.

The proposed framework differs fundamentally in structure:
- It reduces the search space before any evaluation

- It eliminates infeasible goals instead of scoring all options
- It may use simple heuristics only after reduction

Optimization-based characteristics:
- Global evaluation of alternatives
- Dependence on objective function design
- Computational cost scales with search space size

Proposed framework:
- Early elimination of infeasible candidates
- No requirement for global objective formulation
- Bounded computational cost due to reduced space

If optimization is applied, it is restricted to the admissible goal set G_valid, making it localized and computationally tractable.

Structural Differences

The distinction between the approaches can be summarized structurally:
- **GOAP:**
  goal → search → action sequence
- **Reinforcement Learning:**
  state → reward → policy update → action
- **Optimization-based methods:**
  alternatives → objective evaluation → optimal selection
- **Proposed framework:**
  context → constraints → elimination → admissible goals → selection → planning

This formulation emphasizes reduction rather than evaluation.

Key Properties of the Proposed Framework

The proposed approach introduces the following properties:
- **Determinism**
  decisions follow from constraint satisfaction without randomness
- **Interpretability**
  eliminated goals can be traced to specific constraint violations
- **Bounded complexity**
  computational effort is limited by the size of the admissible set
- **Compatibility**
  integrates with existing planning systems such as GOAP

Summary

The proposed framework does not replace existing methods but complements them by addressing the goal selection stage. It:
- Precedes planning by defining admissible goals
- Avoids reliance on stochastic exploration or global optimization
- Ensures consistency between selected goals and system constraints

As a result, it provides a structured and deterministic basis for autonomous decision-making. The framework shifts decision-making from evaluating all possibilities to eliminating those that cannot be sustained.

Illustrative Examples: Role-Dependent Goal Selection Under Minimal Supervision

This section demonstrates the proposed framework using two agents with distinct identity libraries: a **dancer agent** and a **soldier agent**. Both operate under minimal supervision, relying on internal identity constraints and context-driven elimination to determine goals, which are then passed to a planning module such as Goal-Oriented Action Planning.

Two autonomous agents operating under the proposed framework independently determine their goals based on their respective identity libraries and the current environmental context, without external supervision. Each agent maintains a distinct set of admissible objectives and persistent constraints reflecting its operational role and responsibilities. As environmental conditions evolve, context-derived constraints eliminate infeasible goals, ensuring that each agent selects only those objectives that remain consistent with both its internal structure and its interaction with other agents. This process enables coordinated behavior without centralized control, as decisions are shaped by shared environmental constraints and role-specific limitations, resulting in context-adaptive, responsibility-aware goal selection.

Dancer Agent
Identity Library

Goal set:
- G_1: perform choreography
- G_2: adapt movement to environment
- G_3: maintain balance and posture
- G_4: pause or stabilize

Persistent constraints (Π):

- Balance must be maintained
- Collision with environment must be avoided
- Motion must remain within physical limits

Case A: Stable stage, normal conditions

Context K:

- Flat surface
- No obstacles
- Sufficient energy

Constraints C(K):

- minimal restrictions

Result:

G_valid = $\{g_1, g_2, g_3\}$

Residual resolution:

- priority → choreography dominates

Selected goal: **perform choreography (g_1).** GOAP generates motion sequence

Case B: Slippery surface

Context K:

- Low friction detected

Constraints C(K):

- eliminate high-risk movements

Result:

G_valid = $\{g_2, g_3\}$

Residual resolution:

- balance prioritized

Selected goal: **maintain balance (g_3)**

Case C: Near loss of stability

Context K:

- Instability detected

Constraints:

- eliminate all non-stabilizing actions

Result:

G_valid = $\{g_4\}$

Selected goal: **pause or stabilize (g_4)**

Soldier Agent

Identity Library

Goal set:

- G_1: advance
- G_2: hold position
- G_3: retreat
- G_4: seek cover

Persistent constraints (Π):

- Self-preservation must be maintained
- Engagement must remain within operational rules
- Coordination with other agents must be preserved

Case A: Low threat, mission active

Context K:

- Low threat level
- Sufficient resources

Constraints:

- minimal filtering

Result:

G_valid = $\{g_1, g_2\}$

Residual resolution:

- mission priority favors advance

Selected goal: advance (g_1)

Case B: High threat condition detected

Context K:

- incoming threat

Constraints:

- eliminate exposure-increasing goals

Result:

G_valid = $\{g_3, g_4\}$

Residual resolution:

- immediate protection prioritized

Selected goal: **seek cover (g_4)**

Case C: Critical condition

Context K:

- low energy or damage

Constraints:

- eliminate all non-survival actions

Result:

G_valid = $\{g_3\}$

Selected goal: **retreat (g_3)**

Key Observations

Across both agents:

- Goals are not predefined externally

- No stochastic selection is used
- No global optimization is required

Instead:
- Identity defines the goal space
- Context defines constraints
- Elimination reduces the space
- Selection occurs within admissible goals

The agents operate with **minimal supervision**, as all admissibility conditions are embedded internally. This results in:
- Increased autonomy
- Predictable behavior
- Reduced computational burden

Summary
The dancer and soldier agents demonstrate that:
- The same framework applies across distinct roles
- Behavior adapts through constraint variation, not rule rewriting
- Responsibility shifts from external control to internal structure

Different roles produce different behaviors, but the selection mechanism remains identical.

Dynamic Action Libraries and Robust Constraint Filtering for Autonomous Agents

In practical deployments, agents do not operate under continuous external command, but rely on locally available information and internal decision processes. Coordination may arise with minimal or no supervision, where group-level behavior emerges from interaction patterns among agents. This is consistent with swarm-like dynamics, in which collective outcomes are produced without centralized control, yet remain coherent under shared constraints.

To support such behavior, agents maintain a dynamic library of candidate actions. This library is not static; it evolves over time through accumulated observations, prior executions, and interaction outcomes. Let L(t) denote the action library at time t, updated according to:

$$L(t+1) = U(L(t), H(t))$$

where H(t) represents prior interactions and observed outcomes. This update process enables continuous adaptation while preserving continuity with past experience, allowing the system to remain responsive under changing environmental and operational conditions.

Although the library may expand or adapt, it is not unbounded. Its evolution is constrained by the agent's objective and governing principles, ensuring that candidate actions remain consistent with the intended role and operational limits of the system. These principles act as invariant constraints that define acceptable behavior and prevent arbitrary or unsafe actions. Within this formulation, decision-making proceeds as constraint-consistent elimination over the current action library L(t). The system continuously refines its candidate set by filtering infeasible or inconsistent actions, maintaining a balance between adaptability and stability. As a result, agents are able to operate under varying levels of supervision, including fully autonomous conditions, while preserving coordinated and goal-consistent behavior across both individual and group levels [8-9].

A robustness layer evaluates external inputs and influences before they enter the admissible action space. External directives, environmental pressures, or adversarial signals are not assumed to be valid by default; they are subjected to constraint-consistent elimination. Actions that violate safety, integrity, or ethical constraints are rejected at this stage. In addition, the system enforces a viability condition to ensure sustained operation under dynamic conditions. Any candidate action that compromises stability or operational continuity is eliminated, preventing system-level failure while preserving autonomous decision-making.

The absence of continuous supervision does not imply unrestricted behavior or unconditional compliance with external directives. Agents do not execute all possible or suggested actions; instead, decisions are bounded by constraint-consistent evaluation of admissibility. In this sense, behavior reflects a form of system-level common sense, grounded in the preservation of stability, coherence, and operational continuity. At the collective level, this extends to a survival-oriented coordination principle, where agents implicitly act to maintain both individual integrity and group-level viability. As a result, even under minimal supervision, the system avoids arbitrary responses and unsafe compliance, sustaining coherent and context-appropriate behavior.

In contrast to GOAP, which relies on explicit goal selection and plan evaluation, the proposed framework achieves behavior through constraint-consistent elimination, where admissible actions emerge from a dynamically constrained space preserving both individual integrity and collective viability. A persistent self-check mechanism is embedded within the decision loop to ensure internal consistency and prevent drift from governing principles. This mechanism continuously evaluates candidate actions not only against external constraints but also against the agent's own recent behavior, state trajectory, and operational intent. Formally, a consistency operator C(t) acts on the admissible set, such that:

$$A^*(t) = C(t)\,(A(t))$$

where $A(t) \subseteq L(t)$ denotes the constraint-filtered action set. The operator C(t) enforces self-consistency, rejecting actions that introduce internal contradiction, unstable state transitions, or deviation from the agent's functional role. This internal regulation layer enables self-control under autonomy, ensuring that adaptation does not lead to incoherence or unintended behavior. As a result, the system maintains continuity of operation and identity even in the absence of supervision or in the presence of conflicting external inputs. To further strengthen the self-check mechanism, the system may incorporate structured categories of internal evaluation.

These include consistency with prior behavior (history), alignment with the agent's functional role (identity), compatibility with group-level objectives (societal context), and preservation of long-term operational continuity (collective viability). In addition, a notion of internal confidence or self-estimation may be maintained to prevent overreaction to transient inputs or incomplete information. These evaluation dimensions do not introduce new goals but act as stabilizing references, ensuring that decisions remain coherent not only in the immediate context but also across time, interaction, and collective operation. As a result, the system maintains a form of grounded judgment, where action selection reflects continuity, responsibility, and sustained functionality rather than short-term reaction.

This perspective is consistent with behavior-based control and bounded rationality frameworks, where decision-making emerges through layered filtering rather than exhaustive evaluation [10] while differing from GOAP-style approaches that rely on explicit goal selection and plan evaluation [11-12].

Figure 2 illustrates the pre-selection process, where candidate actions are filtered through constraint-based and self-check mechanisms before entering the final task selection stage.

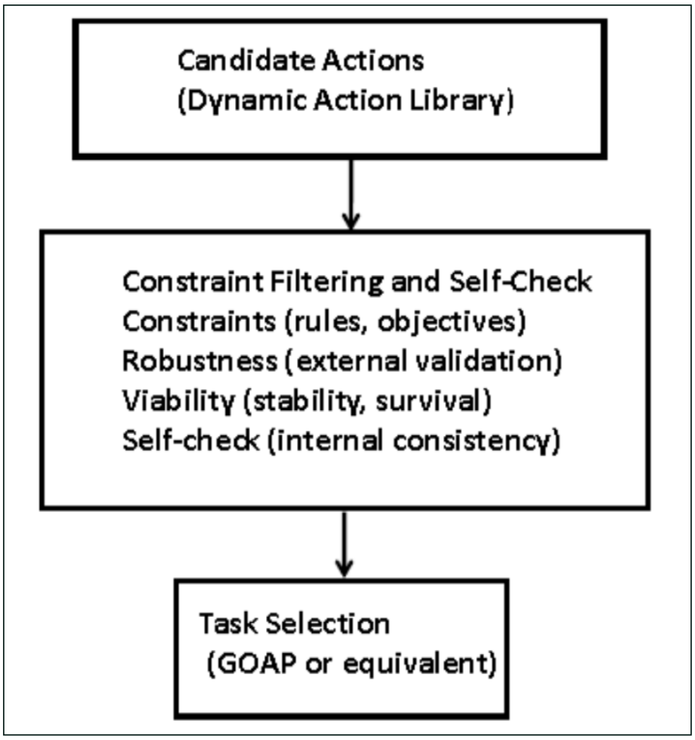

Figure 2: Pre-selection filtering and self-check mechanism. Candidate actions are progressively refined through constraint filtering, robustness and viability checks, and internal self-consistency evaluation before entering the final task selection stage.

The introduction of adaptive behavioral libraries extends the framework to settings where supervision is limited or absent. By allowing the action space to evolve while remaining structurally constrained, the system maintains both flexibility and coherence. This supports swarm-like coordination and reinforces that decision-making can emerge from constraint-consistent elimination over a dynamically updated, yet bounded, set of candidate actions.

The effective constraint set is formed through the combination of identity-driven and context-induced constraints. Persistent identity constraints define the invariant boundaries of admissible behavior, while context-derived constraints introduce dynamic restrictions based on current conditions. Their conjunction, $\Pi \wedge C(K)$, establishes the effective constraint set that governs goal admissibility at each time step. This formulation does not introduce new objectives but reduces the existing goal space by eliminating infeasible candidates. As a result, goal selection emerges as a deterministic consequence of constraint-consistent reduction rather than evaluation or optimization.

Conclusion

This work introduces a deterministic formulation of goal selection in which objectives are not assumed a priori but are obtained through constraint-driven elimination. By separating the definition of admissible goals from the planning of actions, the framework addresses a structural gap in common decision pipelines. An internal identity layer provides a bounded goal library and persistent constraints, while context is mapped to additional constraints that remove infeasible goals. The resulting admissible set defines the space within which a final goal is resolved.

The central shift is from evaluation to reduction. Instead of scoring all alternatives or relying on stochastic exploration, the system eliminates goals that violate identity or context constraints. This produces a bounded, interpretable set of admissible objectives and, in many cases, a unique goal without further processing. When multiple admissible goals remain, residual resolution is performed using deterministic rules over a reduced set, maintaining low computational cost and avoiding global optimization.

The framework is designed to complement, not replace, existing methods. In particular, it operates as a pre-planning layer for systems such as GOAP, supplying a context-consistent goal prior to action planning. Compared with reinforcement learning and optimization-based approaches, the method avoids reward design, stochastic policies, and large-scale search, making it suitable for real-time, resource-constrained, and safety-critical applications where admissibility and predictability are primary requirements.

Future work will focus on (i) formal specification of constraint classes and consistency guarantees for the identity library, (ii) empirical validation through simulation or hardware experiments, and (iii) integration with downstream planners to quantify improvements in robustness and latency. The proposed formulation provides a deterministic and interpretable basis for autonomous decision-making by ensuring that selected goals are consistent with both internal structure and external conditions.

By allowing the action space to evolve while remaining structurally constrained, the system maintains both flexibility and coherence. This supports swarm-like coordination and reinforces that decision-making can emerge from constraint-consistent elimination over a dynamically updated, yet bounded, set of candidate actions. In this formulation, autonomy is realized as freedom within constrained admissibility, ensuring that all selected goals remain consistent with both intrinsic structure and contextual conditions. The framework establishes a deterministic foundation for goal formation, positioning constraint-consistent elimination as a primary mechanism of autonomous decision-making.

Author Contributions

Huseyin Murat Cekirge is the sole author. The author read and approved the final manuscript.

Conflicts of Interest

The author declares no conflicts of interest.

References

1. Sutton RS, Barto AG (2018) Reinforcement Learning: An Introduction. MIT Presshttps://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf.
2. Mnih V, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, et al. (2013) Playing Atari with Deep Reinforcement Learning https://arxiv.org/abs/1312.5602.
3. Boyd S, Vandenberghe L (2004) Convex Optimization. Cambridge University Press https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf.
4. Nocedal J, Wright SJ (2006) Numerical Optimization. Springer https://www.ime.unicamp.br/~pulino/MT404/TextosOnline/NocedalJ.pdf.
5. Russell S, Norvig P (2020) Artificial Intelligence: A Modern Approach. Pearson https://aima.cs.berkeley.edu/.
6. Ghallab M, Nau D, Traverso P (2016) Automated Planning and Acting. Cambridge University Press. https://assets.cambridge.org/97811070/37274/

frontmatter/9781107037274_frontmatter.pdf.

7. Colledanchise M, Ögren P (2018) Behavior Trees in Robotics and AI. CRC Press https://arxiv.org/abs/1709.00084.
8. Pearl J (1984) Heuristic Search: Theory and Applications. Addison-Wesley https://dl.acm.org/doi/book/10.5555/1875144.
9. LaValle SM (2006) Planning Algorithms. Cambridge University Press https://www.cambridge.org/core/books/planning-algorithms/FC9CC7E67E851E40E3E45D6FE328B768.
10. Brooks RA (1986) A Robust Layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation 2: 14-23.
11. Simon HA (1957) Models of Man: Social and Rational. Wiley https://psycnet.apa.org/record/1958-00363-000.
12. Orkin J (2006) Three States and a Plan: The AI of F.E.A.R. Proceedings of the Game Developers Conference (GDC). https://www.gamedevs.org/uploads/three-states-plan-ai-of-fear.pdf.