Desiging Autonomous Decision Maker Agents for Semi-structured MADM

Iraj Mahdavi
Assistant Professor , Industrial Engineering Department ,Mazandaran University of Science & Technology, Babol, Iran.
irajarash@rediffmail.com
Mohsen Akbarpour Shirazi
Assistant Professor , Faculty of Industrial Engineering ,K. N. Toosi University of Technology ,Tehran, Iran.
akbarpour@kntu.ac.ir
Babak Shirazi
Lecturer, Industrial Engineering Department, Mazandaran University of Science & Technology, Babol, Iran. Email:shirazi_b@yahoo.com

Keywords: Semi-structured MADM, Autonomous Decision Maker Agent, Multi-Agent, Decision Surface, Stepwise Response Generator, Feasibility Analyzer, Dynamic Filtering System.

Abstract

Classical models of decision making and system optimization with multiple criteria and complete Multiple Attribute Decision Making (MADM) matrix, use specific methods that categorized classic MADM patterns .But in some situations, MADM matrix is not distinguished completely at the first stages of decision making phase, because of the complexity of environment and uncontrollable variables. These complexities would lead to incomplete cognition and non-optimal decision making. In fact in this type of decision making, complete MADM matrix is vague from cognition range of Decision Maker (DM) .We called these forms “semi-structured MADM”. In semi-structured environment, due to its high degree of complexity, the whole environment is not identifiable for DM. Therefore, decision analysis would be complicated. We design autonomous agents for semi-structured MADM that solves problems when alternatives have incomplete structure and DM is not able to recognize the whole alternatives of the environment for optimal decision making. The model which has been proposed is a systematic approach for semi-structured MADM with multi-layer mathematical model. Each layer core constructed form OR rules and helps to recognize environment sequentially. The agent’s Stepwise Response Generator moves in semi-structured environment over decision surface step by step to generate hidden alternatives that DM was unable to recognize them. The new alternatives follow Feasibility Analyzer and Dynamic Filter Module. The procedure is continued with a closed loop feedback which results in construction of the Meta-Decision phase.

1. Introduction

In recent decades, the classic decision making and optimization with one criterion or one objective function has emerged to Multiple Criteria Decision Making (MCDM) model for complex decision making. These models can be linear, nonlinear, or mix models. In these decision makings, utilizing multiple criteria, instead of one criterion is possible. Two categories for decision making can be proposed as Multiple Objective Decision Making (MODM), and, MADM [1,2]. Classic MODM is for planning and classic MADM is for selecting best alternative. Full-structured MADM model can be formulated as decision making matrix as shown in Table1, which DM must select best alternative or best order of alternatives:

Table 1 Full-structured static MADM matrix

$$
D = \begin{array}{c|cccccc|}
 & x_1 & x_2 & . & . & . & x_n \\
\hline
A_1 & a_{11} & a_{12} & . & . & . & a_{1n} \\
A_2 & a_{21} & a_{22} & . & . & . & a_{2\,n} \\
. & . & & & & & \\
. & & & & & & \\
. & & & & & & \\
A_m & a_{m1} & a_{m2} & . & . & . & a_{mn}
\end{array}
$$

$A_1, A_2, \ldots, A_m$ in the matrix D is predefined alternatives and $x_1, x_2, \ldots, x_n$ present attribute utility for any alternative, where a_{ij} elements present special value for j-th attribute and i-th alternative. All considered classic technique for
decision making is related to time when the matrix is deterministic and specific. But if problem does not have a clear structure, due to environment complexities, we would not be able to construct matrix D completely. These
partial observable domains could have static or stochastic dynamic classification [3,4,5,6] .So we can use MADM at different structure depend on type of problem as shown in Figure 1.

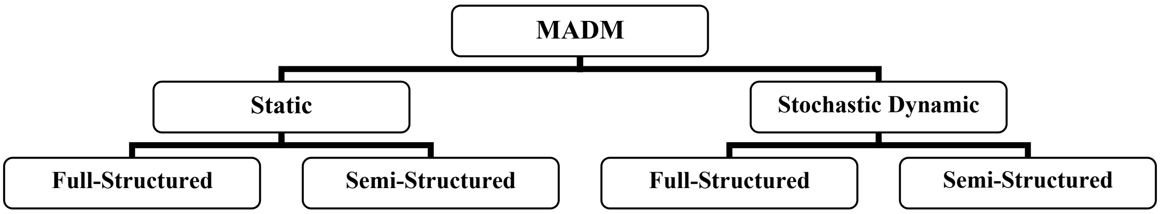


Figure 1 MADM classification

In this paper we consider on static structures. In static MADM system, decision making does not depend on time and matrix has non-dynamic structure. But it can be incomplete which is called "semi-structured MADM ". If decision making system was full-structured static MADM, then we could use the following decision making techniques as shown in Figure 2 [1,2]:

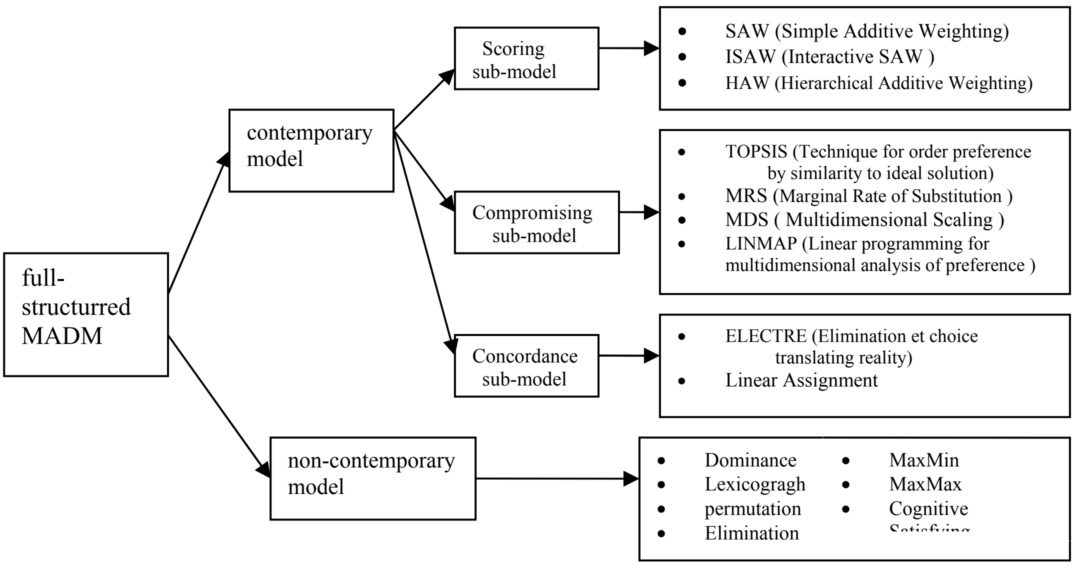


Figure 2 Methods for solving Full-structured static MADM

All above methods of classic MADM can be used when deterministic and specific mathematics model are available. In these cases all system variables are determined. There are no uncontrollable variables in these structures and decision making is in the form of deterministic and, full option of MADM matrix. In contrast, the complexity of problem about alternative recognizing causes to incomplete recognizing to system and consequently to incomplete decision making [4]. These complexities and followed decision could lead to decision bottleneck of the system. Complexities can be formed in mutual relation through « DM- Environment » just because decision making process involves human nature from one point of view and environment from another point of view [7]. These complexities include the following item:

1-DM intelligence: Inability in problem recognizing, inability in problem definition, inability of problem prioritization, inability in information organizing.

2-Problem planning: Inability to generate alternatives, inability in explaining and evaluating.

3-Selecting: Inability in selecting solution method, inability in selecting alternatives.

4-Counting: Large number of alternatives.

5-Environment : Environment with uncontrollable variables, environment disturbance.

These factors lead to form "environment with high degree of complexity" for decision making. In fact in these cases a part of MADM matrix, can be invisible from view of vision. These structures have been called **"semi-structured"** decision model as given in Table2. Alternatives A_{m+j} , which $j \geq 1$, are invisible at first.

Table 2 semi-structured MADM matrix

	x_1	x_2	.	.	.	x_n
A_1	a_{11}	a_{12}	.	.	.	a_{1n}
A_2	a_{21}	a_{22}	.	.	.	a_{2n}
.						
.						
A_m	a_{m1}	a_{m2}	.	.	.	a_{mn}
A_{m+1}	✗	✗	.	.	.	✗
.	✗	✗				✗
.						
A_{m+j}	✗	✗				✗
.	✗	✗				✗
.	✗	✗				✗

Because of invisibility in these incomplete structures, the analyst's task is to help DM for making better decision. In semi-structured situation, incremental exploration of alternatives helps in recognizing the environment. Such a model is used when alternatives involve incomplete structures. It means some alternatives can be recognized by DM and some others have not been seen for the reasons above. This proposed model should be in a way that decrease "decision bottleneck ". For decision making in this environment we design an autonomous decision maker agent. These agents consider the incomplete structure of environment, and begin to recognize environment step by step and generate unknown alternatives of system and provide feedback to reform the alternative plan to complete the matrix structure. It means the agents determine the unspecific portion of system for DM and help him in decision making process. Therefore, we design multi-layer architecture of agent and basic rule of each layer. Each layer core is based on OR rules for designing each module.

2. System behavior analysis with high-level descriptive functions

We know that a semi-structured environment is not recognized by DM completely because it does not have clear structure, so for analyzing, step by step recognizing of system and synthesizing a decision, we need descriptive functions which describe high level behavior of system .We analyze the system as hierarchical structure. For achieving such structure, we should extract attributes of decision system at first $x_1,x_2,...,x_n$ [8,9].Sub-goals of the system could be multiple and complex and could help us for describing the behavior of the system. The hierarchical diagram could be designed for describing of the behavior of system. This structure is based on 3 layers as Goal, Sub-goal and attributes as shown in Figure 3 [10].

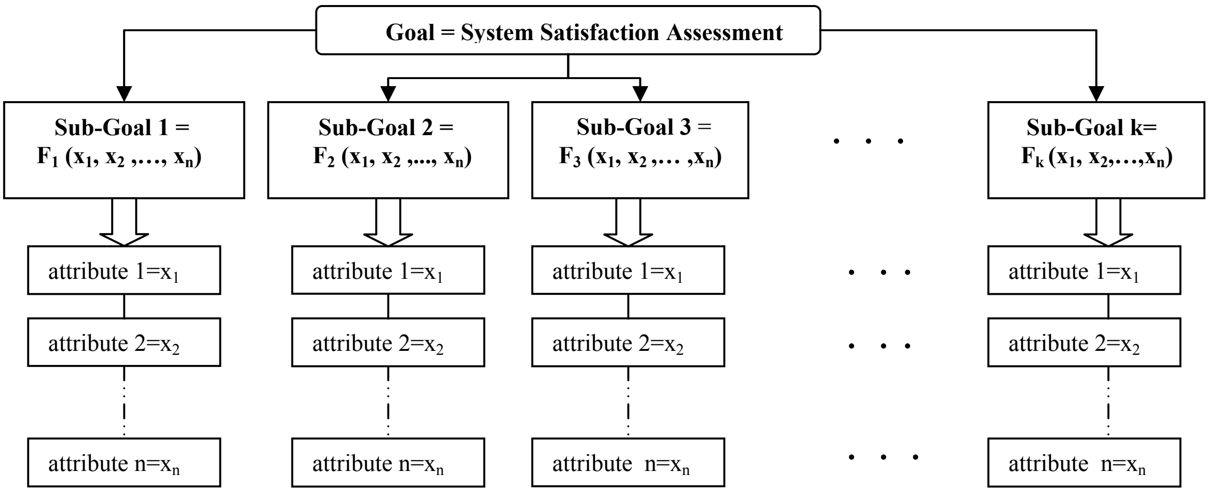


Figure 3 System high-level descriptive functions

Each sub-goal is a function of extracted attributes of decision system that describes system high-level behavior. These functions should design based on past experience of system (DM's believes, analyst believes, past experience of system, past activity of system) that naturally possess nonlinear nature [11].We can use the regression method in order to fit on function in an n-dimensional Euclidean space. Some high level functions that describe system behavior (system sub-goals according the above model) are Stability, Productivity, Performance and Flexibility, that they are considered as function of n-attributes of system:

Subgoal1 =Stability =$F_1(x_1, x_2,…, x_n)$*; Subgoal2 = Productivity=* $F_2(x_1, x_2, …,x_n)$ *;*
Subgoal3=Performance = $F_3(x_1, x_2,…,x_n)$ *; Subgoal4 = Flexibility=* $F_2(x_1, x_2, …,x_n)$ *.*

3. Autonomous Decision Maker Agent

3.1 Decision Making with Autonomous Agent

Plan of decision making in semi-structured environments needs an automaton for searching unknown portion of decision environment. This automaton is for either automated decision making process or as mechanism for generating unknown alternative of system [12]. For this purpose we use Autonomous Decision Maker Agents [13]. An agent is a supporter for DM and act as an interface for reducing complexity of the environment. (We mentioned that semi-structured environment is complex system which under loaded in information).Agent helps DM in decision making process based on initial data that DM has recognized. So as we will see, this mutual relation between « DM-Agent » should always exist as shown in Figure 4.

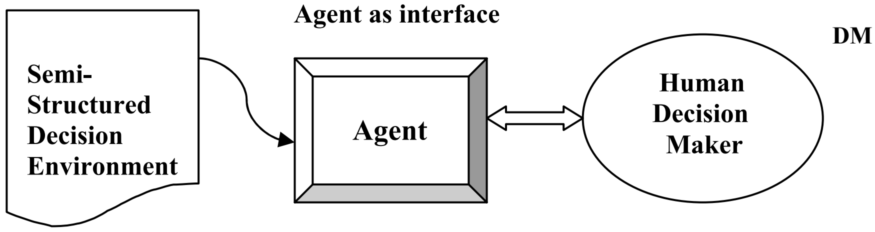


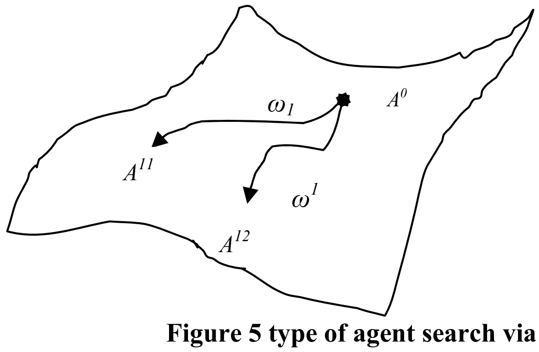
Figure 4 «DM – Agent » mutual relationship

Autonomous agent means generating new alternatives autonomously which DM could not recognize them, independency from other agents in multi-agents system and in developing state as substitute of DM.For reaching final goal of system and due to semi-structured nature of problem, Autonomous agent decision making is sequential (stepwise) [14]. Each time agent receives information from environment can help DM to decision making and in evolutionary manner can be replaced with DM [15]. Table3 shows comparison between Human DM and Autonomous DM Agent [12].

Table 3 Comparison between Human DM and Autonomous DM Agent

Decision Making With DM	Decision Making with Autonomous Agent
Decision making and evaluation with few alternatives	Generates and evaluates many alternatives
Confines to DM experience	Uses data mining and search technique
Uses general rules and few decisions	Uses innovative techniques and Meta-Decision
Difficult to search in vague environment	Searches in vague space and generate alternatives mo
Not systematic approach for solving semi-structured MA	Uses continuous feedback system for revising decisi

Moving on semi-structured environment must have sequential nature [14], so agent should use incremental information. As we will see, in next section, DM agent has a multi-layer structure that search in vague space to generate alternatives. This agent moves over decision surface and presents vague portion of decision system. So this agent is a search agent which is used for improving decision, by local searching on decision surface and varies the search area in search space as shown in Figure 5 [16,17].



$A^0 = (x_1, x_2, \ldots, x_n)$
$A^{11} = (x_1^{\ 11}, x_2^{\ 11}, \ldots, x_n^{\ 11})$
$A^{12} = (x_1^{\ 12}, x_2^{\ 12}, \ldots, x_n^{\ 12})$

Figure 5 type of agent search via path ω_1 or ω^1

A^0 is one of m predefined alternatives $(A_1, A_2, \ldots, A_m)$ and A^{11}, A^{12} are new generated alternatives.

3.2 Designing autonomous agent

Autonomous agent for decision making in semi-structured with high degree of complexity should use method which considers vague structure properties and sequentially help to move on unspecified decision making area. This agent should be structured as given in Figure 6:

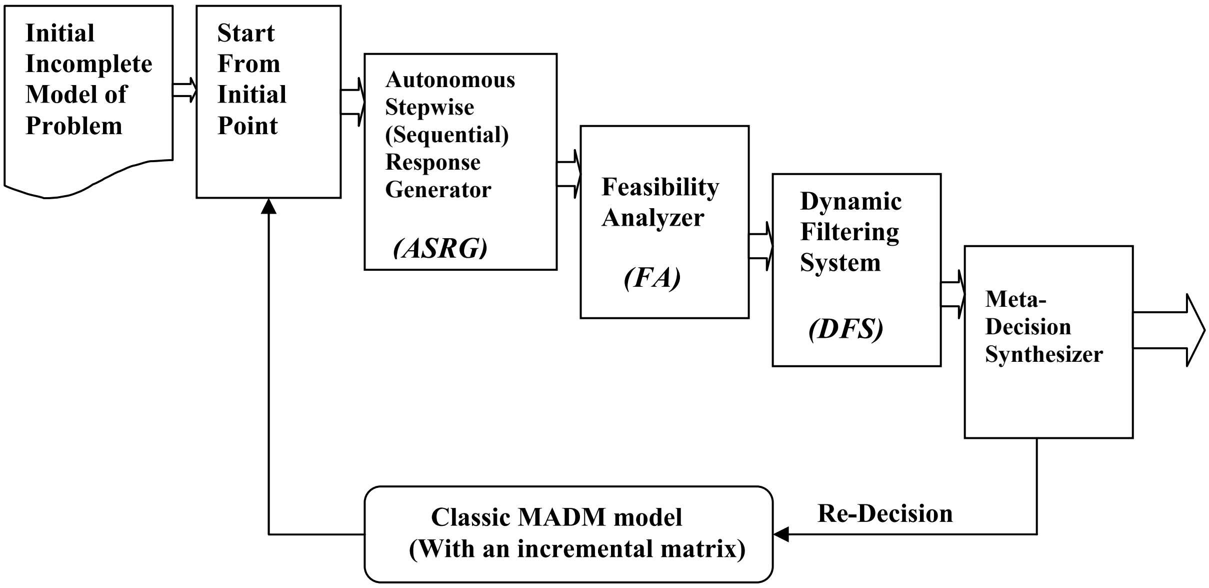


Figure 6 Autonomous Decision Maker Agent Architecture

In the following, we present general concept about each module:

Module 1- Start from an initial point: Initial point from decision surface for moving.

Module 2- Autonomous Stepwise (Sequential) Response Generator (ASRG): Autonomously generating new responses from initial point step by step.

Module 3- Feasibility Analyzer (FA): Feasibility of generated alternative in the space of problem solution.

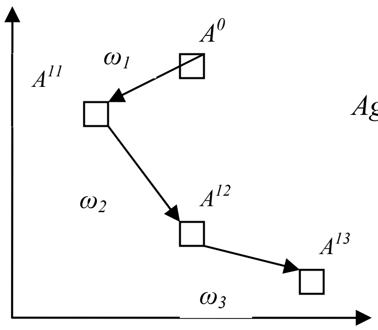
Module 4-Dynamic Filtering System (DFS): Presentation of new alternative for DM and elimination irrelevant alternatives based on interaction with DM or emphasis on the selection of some alternative which includes some specific condition.

Module 5- Meta-Decision Synthesizer: Extracting effective alternatives and add them to initial matrix.

Re-Decision: Reproduction MADM matrix based on generating of new alternative in an incremental matrix and utilizing classic MADM technique.

4. Decision surface and Autonomous Stepwise (Sequential) Response Generator

The method we use for generation alternative in vague semi-structured environment is based on sequential movement logic, in decision space from predefined points .Because alternative generation is a step by step process with movement on decision surface, hence stepwise response generator is needed. Agent uses “decision surface “for new decision making with movement method on these surfaces as shown in Figure 7.



$$Agent\ : A^0 \rightarrow A^{11} \rightarrow A^{12} \rightarrow A^{13} \ldots \rightarrow A^{1i} \rightarrow \ldots$$

Figure 7 Agent movement and generation new alternatives

The important point is that, the environment complexity is interrelated to movement on decision surface and based on this fact, for the modeling of this surface we can use the regression method in order to fit on function in an n-dimensional Euclidean space. Therefore to solve semi-structured problem, we use decision surface which is considered of initial certain point of system (m predefined alternatives) based on environment complexity and we can assign more compnex function to this surface for more complex environment.

4.1 Definition: decision surface

F is general form of descriptive function and points set as $A = \{x_1, x_2, \ldots, x_n\}$ that satisfy following condition
construct decision surface (A is general form of each alternative) :

$$F: \mathbb{R}^n \rightarrow \mathbb{R}$$
$$(x_1, x_2, \ldots x_n) \in \mathbb{R}^n \quad | \quad x_i \in R \;\Rightarrow\; A \in \mathbb{R}^n$$
$$\Rightarrow F_i(A) \in \mathbb{R}\;,\; i=1,2,\ldots,k \quad (\text{ k descriptive functions })$$

R^c is:

$$R^c = \{A \mid A \in \mathbb{R}^n, F_i(A) \in \mathbb{R}\}$$

Ω is decision surface

$$\text{Decision surface} = \Omega(A) = \{A \mid A \text{ has above conditions}\}$$
$$\Rightarrow \Omega(A) = \Omega(x_1, x_2, \ldots, x_n) = \text{ is supposed one of descriptive functions} = F_i(A)$$

Decision surface is a n-dimensional subspace from a n+1 dimensional space which is divided into R^c and $\mathbb{R}^n - R^c$ .Based on of complexity we can assign several forms for decision surface .

$\Omega(A) = b_0 + \sum b_i x_i \quad 1 \le i \le n$ *(linear decision surface for low complexity)*
$\Omega(A) = b_0 + \sum b_i x_i + \sum \sum b_{ij} x_i x_j \quad 1 \le i \le n \;, 1 \le j \le n$ *(quadratic decision surface for mean complexity)*
$\Omega(A) = b_0 + \sum b_i x_i + \sum \sum b_{ij} x_i x_j + \sum \sum \sum b_{ijk} x_i x_j x_k \quad 1 \le i \le n \;, 1 \le j \le n \;, 1 \le k \le n$ *(higher complex environment)*

Type of decision surface is modeled by a regression model of system descriptive function based on its complexity. Agent begins movement on decision surface ψ from certain point $A_1, A_2, \ldots, A_m$ through the path ω with algorithms φ_h (is described in the next section).Figure 8 shows agent starts its movement from A_1 , and generate A^{11} through path ω_1 over decision surface Ω (A) .

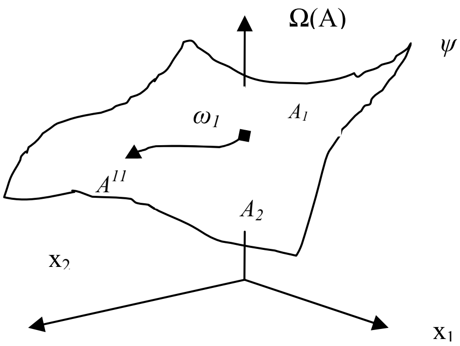


Figure 8 agent movements over decision surface ψ

4.2 Sequential (stepwise) algorithms for moving on decision surface

Suppose that Ω *(A)* is decision surface, agent starts from selected and applicable point A_k on decision surface. ξ_k is amount of step size and its selection method is important because if it is too small, the trend of agent movement is too slow and if it is too large the system will face overshooting.

4.2.1 Algorithm φ_h*-1:* $(A_{k+1} = A_k - \xi_k \; \partial\Omega(A_k))$

1- Begin

2- initialize A, σ, ξ , k=0
3- do k ← k+1
4- A ← A – ξ_k ∂Ω(A)
5- until ξ_k ∂Ω(A) < σ
6- return A
7- End.
(A_{k+1} : new generated alternative)

We now consider a principled method for setting ξ_k. Suppose that the function can be well approximated by the second-order expansion around a value A_k :

$$\Omega(A) \approx \Omega(A_k) + \partial \Omega^t (A - A_k) + \tfrac{1}{2} (A - A_k)^t H (A - A_k)$$

Where H is Hessian matrix of second partial derivation $\partial^2 \Omega / \partial A_i \partial A_j$ evaluated at A_k.Then according to the above relation:

$$\Omega(A_{k+1}) \approx \Omega(A_k) - \xi_k || \partial \Omega ||^2 + \tfrac{1}{2} \xi_k^2 \partial \Omega^t H \partial \Omega$$

That after minimization

$$\xi_k = || \partial \Omega ||^2 / \partial \Omega^t H \partial \Omega$$

To minimize the second-order expansion, we can also use the following algorithm:

4.2.2 Algorithm φ_h-2 : ($A_{k+1} = A_k - H^{-1} \partial \Omega$)

1- Begin
2- initialize A, σ, k=0
3- do k ← k+1
4- A ← A – H^{-1} ∂ Ω(A)
5- until H^{-1} ∂ Ω(A) < σ
6- return A
7- End.

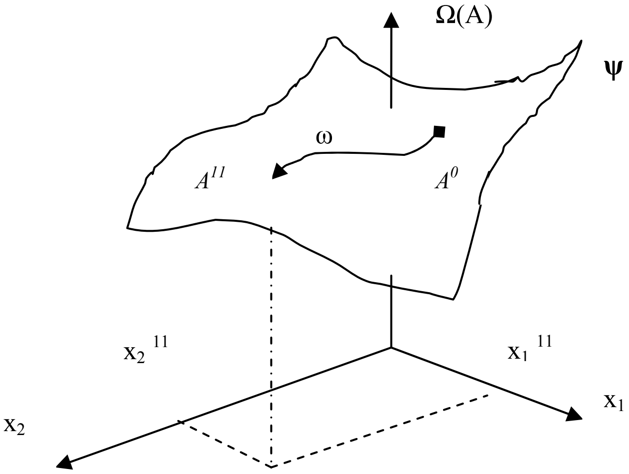
In simple term, we can use following relation:

$$\Rightarrow \quad A_{k+1} = A_k - \omega_k \xi_k$$

ξ_k is step size and ω_k is search direction (path ω) over decision surface.

5. Feasibility Analyzer

Due to generation of new alternatives over decision surface in semi-structured environment, it is possible some new generated alternatives, in some attributes x_i does not satisfy $x_i^L \le x_i \le x_i^R$. So we use a Feasibility Analyzer for considering of feasible solution of new generated alternatives. New alternative A is feasible if all of its attributes are in the feasible region ($x_i^L \le x_i \le x_i^R$; for any i ,1 ≤i ≤ n).Over the decision surface **ψ,** new alternative A^{11} is generated from path ω. For feasibility of A^{11} as shown in Figure 9 we consider the following constraint:



Feasibility Analyze of A^{11} :

1) $x_1^L \le x^{11}{}_1 \le x_1^R$

2) $x_2^L \le x^{11}{}_2 \le x_2^R$

Figure 9 Extracting new alternative attributes

Constraints $x_i^L \le x_i \le x_i^R$, in $n+1$ dimensional space (due to n attribute) of problem, construct a subspace that may intersect decision surface.

6. Dynamic Filtering System

Designing filter is multi-stage, repetitive activity in interaction with DM. Filter has dynamic nature due to changing of DM priority and attitude. However filtering specializes some points on the new points in order to eliminate additional points and some certain points will remain in order to assert DM attitude framework, or inversely the emphasis is on remaining alternative with special frame in DM point of view which they remain in decision system region after filtering.

6.1 General Filters:

If A^- and A^+ are arguments of min and max limits of descriptive function F that is attitude of DM:

$$F(A^-) = \min F(A) \quad , \quad F(A^+) = \max F(A)$$

Then we can design three general forms of filters as shown in Figure 10:

a) Low–Pass Filter: Eliminate all alternative that $F(A) \ge F(A^+)$ or pass all alternatives that $F(A) < F(A^+)$

b) Middle–Pass Filter: Eliminate all alternative that didn't satisfy $F(A^-) \le F(A) \le F(A^+)$ or pass all alternatives that $F(A^-) < F(A) < F(A^+)$

c) High–Pass Filter: Eliminate all alternative that $F(A) \le F(A^-)$ or pass all alternatives that $F(A) > F(A^-)$

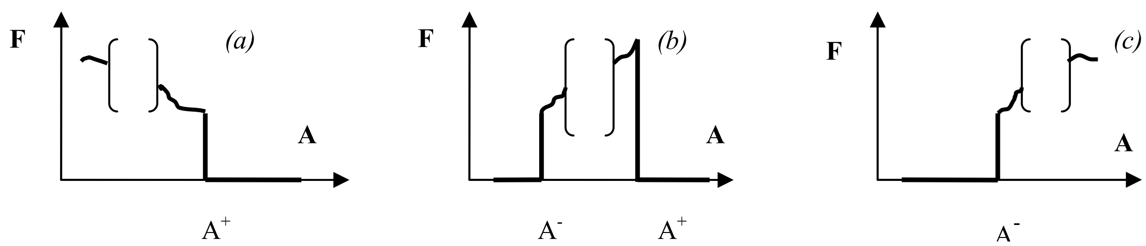


Figure 10 Low-Pass-Filter, Middle-Pass-Filter and High-Pass-Filter (left to right)

These are general forms of filtering systems. Some of alternatives could be eliminated as a filter is applied. Filtering new generated alternatives has a dynamic nature because of they are generated autonomously by agent,

out of DM control, so DM attitudes should be applied. Filters may have different structure compare to above
general filters, but they should eliminate some unnecessary alternative. So we can consider different type of filter according to DM's comment.

6.2 Similarity Filter

The filtering system for the elimination of new generated points which are close to each other very much:

$$\{ \sum [\gamma_l \,|\, F_l (A^t) - F_l (A^h) \,|^p] \}^{1/p} \le \delta \quad , \qquad l = 1,2,\ldots,k \text{ (k descriptive functions)}$$

F_l system descriptive function, γ_l priority of descriptive function

A^t new generated alternative that should be considered for non-similarity

A^h alternative that passes through filter from last consideration and remain in current set

δ is adjusting parameter for non-similarity of point , $1 \le p \le \infty$

Suppose effective point A^h as initial point with δ and pass other (remain) points form filter. Each point that has distance less than δ should be eliminated .Hence initial set decrease to smaller set .We select point of new set that have smallest distance of A^h . This point is second base point of comparison in filter and then repeat above process. We continue above steps until an effective subset of decision points remain.

7. Meta-decision synthesizer and Re-Decision

Definition: meta-decision

The movement on decision surface ψ from certain point $A_1, A_2, \ldots, A_m$ through the path ω with algorithms φ_h would cause to generate new alternatives which DM couldn't identify them from the beginning. These represented alternatives which are added to the first decision matrix will form another new matrix which can lead to new decision. Besides responses generated by classic technique from initial matrix, agent adds new alternatives to decision matrix as shown in Figure 11 which can make more exact decision, and we called this process "meta-decision".

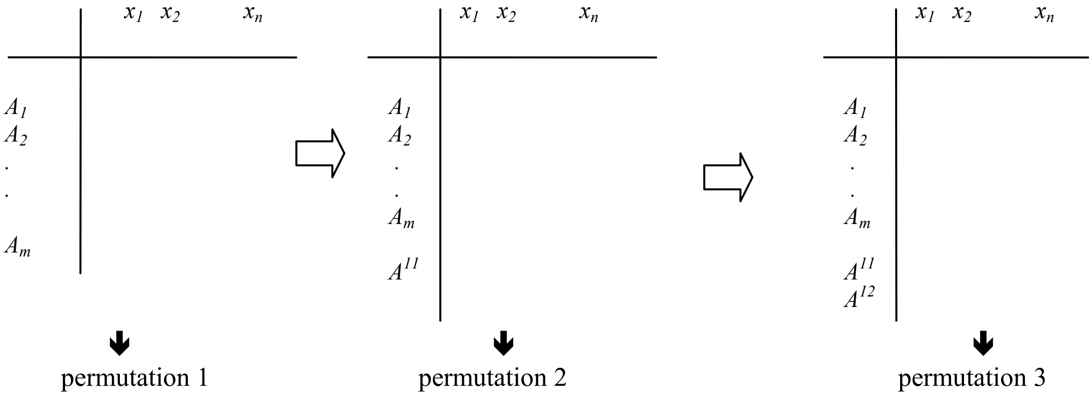


Figure 11 Meta-Decision

New generated alternatives from stepwise generator that pass through Feasibility Analyzer and Dynamic Filtering System, can add to initial decision system. In this new generated matrix, we can perform according to classic MADM technique and make decision again (Re-Decision).Therefore a different output could result with repetition each time (Meta-Decision)as shown in Figure 12, and their outputs can lead to an effective and flexible selection by interactive with the DM.

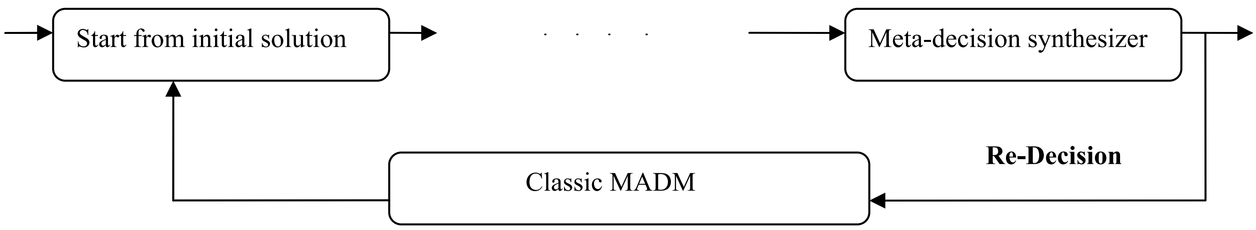


Figure 12 Meta-Decision over Re-Decision phase

8. Multi-Agent Architecture

8.1 Multi-agent and semi-structured

Any agent who has the above mentioned structure as autonomous decision maker agent can support DM in decision making. But, for a faster process in environment with higher degree of complexity multi agent architecture can be used [18]. Each agent autonomously will start searching decision surface regarding high level goal, independent of the other agents, with an initial structure of decision for all agent. Each would have an independent response which might be incompatible with the other. But the most important concept in Multi-agent is final "collaboration" and "cooperation" among agents for fulfillment of collective agreed point [19]. To achieve this goal agents should compromise together. Here by the interacting with DM ,the problem can be solved by using one of the solution which agents proposes to DM as shown in Figure 13.

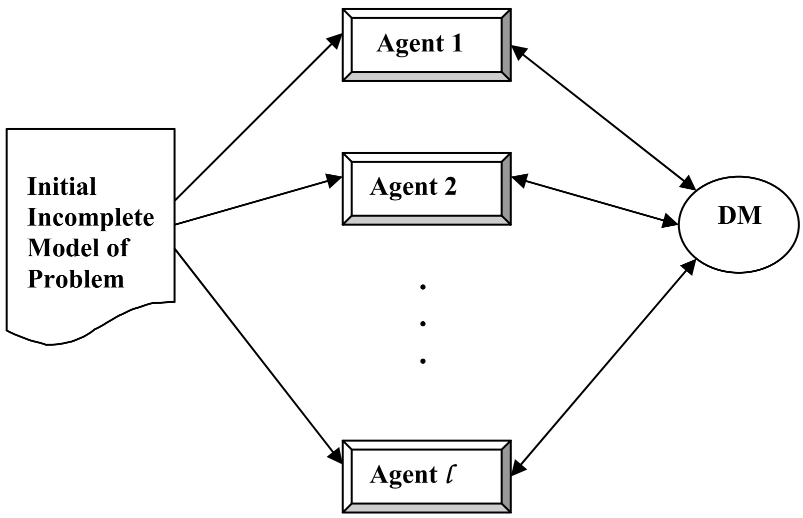


Figure 13 Multi-Agent system for decision making in interaction with DM

8.2 An algorithm for Multi-agents in semi-structured environment

Suppose that in a system there are l agents (Agent 1,Agent 2,…, Agent l), and each of these agents move on decision surface and generate new alternatives with algorithm φ_h (this algorithm is used step by step moving on decision surface to generate alternatives that will be described).

We define function $P_j\ (A^{ji})$ for each agent at any point of A^{ji} corresponding to *j-th* agent at *i-th* generated point as follows:

Agent 1 : $A^0 \rightarrow A^{11} \rightarrow A^{12} \rightarrow \ldots \rightarrow A^{1i} \rightarrow \ldots$
Agent 2 : $A^0 \rightarrow A^{21} \rightarrow A^{22} \rightarrow \ldots \rightarrow A^{2i} \rightarrow \ldots$
.
.

i-th repetition of all agents

$$Agent\ l:\ A^0 \rightarrow A^{l1} \rightarrow A^{l2} \rightarrow \dots \rightarrow A^{li} \rightarrow \dots$$

$A^{ji} = \varphi_h(A^0)$ is new point on decision surface generated by φ_h algorithm (with agent j , starting from A^0,that A^0 is one of the m predefined alternatives). For each Agent j at A^{ji} we define function $P_j(A)$ as follow:

$$P_j(A^{ji}) = \| F(A^0) - F(A^{ji}) \| \qquad \textit{function } P_j(A^{ji}) \textit{ at point } A^{ji} \textit{ related to } A^0 \textit{ for agent } j$$

We suppose agent j has weight w_j at first ($\sum w_j = 1$).

Stage 1 – any agent from point A^0 starts its movement on decision surface according to the above mentioned trends and the algorithm assigned for movement on the surface.

Stage 2 – the amounts of function $P_j(A^{ji})$ for agent j $(1 \le j \le l)$ in i-th repetition alternative generation :

$$\{ P_1(A^{1i}), P_2(A^{2i}), \dots, P_l(A^{li}) \}$$

Stage 3 – if descriptive function F has positive nature such as stability, performance, productivity… then:

$$A^* = \underset{1 \le j \le l}{\arg\max}\ P_j(A^{ji}) \quad (\textit{at i-th repetition}) \quad \rightarrow \quad \text{Agent with best alternative}$$

$$A_* = \underset{1 \le j \le l}{\arg\min}\ P_j(A^{ji}) \quad (\textit{at i-th repetition}) \quad \rightarrow \quad \text{Agent with worst alternative}$$

note : $\arg\max P_j(A^{ji}) = \arg\max \{ P_1(A^{1i}), P_2(A^{2i}), \dots, P_l(A^{li}) \} = \arg P_k(A^{ki}) = A^{ki} \rightarrow$ *k-th agent*

else if function F has negative nature such as cost :

$$A^* = \underset{1 \le j \le l}{\arg\min}\ P_j(A^{ji}) \quad (\textit{at i-th repetition}) \quad \rightarrow \quad \text{Agent with best alternative}$$

$$A_* = \underset{1 \le j \le l}{\arg\max}\ P_j(A^{ji}) \quad (\textit{at i-th repetition}) \quad \rightarrow \quad \text{Agent with worst alternative}$$

Stage 4 – we increase (decrease) the weigh of best (worst) agent up (down) to δ.

Stage 5 – $A^0 = A^*$ and then return to the fist stage .

9. Numerical example

8.1 problem description

Table 4 shows a system with 3 attributes in which DM recognizes three alternatives. If one of each alternative is selected, then corresponding stability achieved for the system.
(x_1 , x_3 has positive nature and x_2 has negative nature)

Table 4 Stability of each alternative

x_1^+	x_2^-	x_3^+		System performance
7	average	6	→	34%
9	low	4	→	25%
6	high	7	→	46%

Due to complexity of environment, DM can't recognize other alternatives of the system, and then we use an autonomous agent to generate alternatives. First we construct semi-structured matrix (use "norm technique "to achieve dimensionless matrix) as given in Table5:

Table 5 Constructing semi-structured matrix from initial incomplete MADM matrix

	x_1^+	x_2^-	x_3^+
A_1	7	5	6
A_2	9	3	4
A_3	6	7	7

$\Rightarrow$

	x_1^+	x_2^-	x_3^+
A_1	0.54	0.55	0.6
A_2	0.7	0.33	0.4
A_3	0.46	0.77	0.7

With applying TOPSIS to this matrix, we achieve following permutation:

TOPSIS ➔ $A_2 > A_1 > A_3$

8.2 using Autonomous agent to make decision more exact:

We assign a quadric form of decision surface to above alternative, according to regression model of stability function:

$$\textit{Stability Function of System} = F_3\,(x_1, x_2, x_3) = \Omega\,(A)$$
$$\Rightarrow \Omega\,(A) = x_1^2 + 2\,x_1\,x_2 + 2x_2^2 + x_1 + 2x_2\,x_3 + 2x_3^2$$
$$\Rightarrow \partial\Omega\,(A) = \{\, \partial\Omega(A)/\partial x_1\,,\ \partial\Omega(A)/\partial x_2\,,\ \partial\Omega(A)/\partial x_3 \,\} =$$
$$= \{\ 2\,x_1 + 2\,x_2 + 1\,,\ 2\,x_1 + 4\,x_2 + 2\,x_3\,,\ 2\,x_2 + 4\,x_3\ \}$$

Agent's ASRG starts to move on decision surface Ω (A) from point A_1 on it. Agent select step size randomly due to acting autonomously:

$$\omega_1 = \partial\,\Omega(A)|_{A=A1}$$
$$\Rightarrow \omega_1 = \{\, 3.9\,,\ 6.2\,,\ 3.5 \,\}$$
$$\xi_1 = 0.05$$
$$A_{k+1} = A_k - \omega_k\,\xi_k$$
$$\Rightarrow A^{11} = A_1 - \xi_1\,\omega_1 = \{\, 0.54\,,\ 0.55\,,\ 0.6 \,\} - 0.05\,\{\, 3.9\,,\ 6.2\,,\ 3.5 \,\} = \{\, 0.345\,,\ 0.24\,, 0.425 \,\}$$
$$\xi_2 = -0.1$$
$$\omega_2 = \partial\,\Omega(A)|_{A=A^{11}}$$
$$\Rightarrow A^{12} = A^{11} - \xi_2\,\omega_2 = \{\, 0.345\,,\ 0.24\,, 0.425 \,\} + 0.1\,\{\, 2.17\,,\ 2.5\,,\ 2.18 \,\} = \{0.562\,,\ 0.49\,, 0.643\,\}$$
$$\xi_3 = -0.1$$
$$\omega_3 = \partial\,\Omega(A)|_{A=A^{12}}$$
$$\Rightarrow A^{13} = A^{12} - \xi_3\,\omega_3 = \{0.562\,,\ 0.49\,, 0.643\,\} + 0.1\{\ 3.1\,,\ 4.36\,,\ 3.55\,\} = \{\, 0.872\,,\ 0.92\,,\ 0.998\,\}$$
$$\xi_4 = 0.1$$
$$\omega_4 = \partial\,\Omega(A)|_{A=A^{13}}$$
$$\Rightarrow A^{14} = A^{13} - \xi_4\,\omega_4 = \{\, 0.872\,,\ 0.92\,,\ 0.998\,\} - 0.1\{\, 4.58\,,\ 0.741\,,\ 5.83\,\} = \{\, 0.414\,,\ 0.18\,,\ 0415\,\}$$

Agent's FA (Feasibility Analyzer) has the following limit on system upper and lower bound of attribute:

$$x_1^L = 0.1 \quad x_1^R = 0.9 \quad \Rightarrow \quad 0.1 \le x_1 \le 0.9$$
$$x_2^L = 0.1 \quad x_2^R = 0.9 \quad \Rightarrow \quad 0.1 \le x_2 \le 0.9$$
$$x_3^L = 0.1 \quad x_3^R = 0.9 \quad \Rightarrow \quad 0.1 \le x_3 \le 0.9$$

Consequently alternative A^{13} is not feasible, and therefore, it is eliminated from list of new generated alternatives. Filtering system of this agent is designed, so that, alternatives with stability less than 20%

can't pass through it. Then a High-Pass-Filter is needed, in which argument A^- produces at least 20% stability. Consequently passing alternatives through Filter, A^{11}, A^{14} are removed from list, because:

Stability of $A^{11} \rightarrow 13\%$
Stability of $A^{14} \rightarrow 12\%$

And finally A^{12} remains and adds to the first matrix as shown in Table6. Incremental matrix can be solved with classic MADM:

Table 6 Incremental matrix of MADM

	x_1^+	x_2^-	x_3^+
A_1	0.54	0.55	0.6
A_2	0.7	0.33	0.4
A_3	0.46	0.77	0.7
A^{12}	0.562	0.49	0.643

With applying TOPSIS to this matrix, we achieve following permutation:

TOPSIS $\rightarrow$ $A^{12} > A_2 > A_1 > A_3$

We continue this process again (because of agent feedback), starting with A_2:

$\xi_5 = 0.05$
$\omega_5 = \partial\, \Omega\,(A)|_{A=A2}$
$\rightarrow A^{21} = A_2 - \xi_5\ \omega_5 = \{ 0.7\ ,\ 0.33\ ,\ 0.4 \} - 0.05 \{ 3.06\ ,\ 3.52\ ,\ 3.26 \} = \{ 0.547\ ,\ 0.154\ ,\ 0.286 \}$
$\xi_6 = -0.1$
$\omega_6 = \partial\, \Omega(A)|_{A=A^{21}}$
$\rightarrow A^{22} = A^{21} - \xi_6\ \omega_6 = \{ 0.547\ ,\ 0.154\ ,\ 0.286 \} + 0.1 \{ 2.402\ ,2.282\ ,1.452 \} = \{0.787,0.383,0.431\}$
$\xi_7 = -0.05$
$\omega_7 = \partial\, \Omega(A)|_{A=A^{22}}$
$\rightarrow A^{23} = A^{22} - \xi_7\ \omega_7 = \{0.787, 0.383, 0.431\} + 0.05 \{3.34, 3.96, 2.49\} = \{0.954, 0.581, 0.551\}$
$\xi_8 = 0.1$
$\omega_8 = \partial\, \Omega(A)|_{A=A^{23}}$
$\rightarrow A^{24} = A^{23} - \xi_8\ \omega_8 = \{0.954, 0.581, 0.551\} - 0.1 \{4.07, 5.33, 3.36\} = \{0.547, 0.048, 0.215\}$

Consequently alternatives A^{23}, A^{24} are not feasible, and they are removed from list of new generated alternatives. A^{23} is infeasible due to x_1 and A^{24} is infeasible due to x_2.
The same filter eliminates A^{21} after passing it. Because of its stability is less than 20%:

Stability of $A^{21} \rightarrow 13\%$

Finally A^{22} remains and adds to the last matrix as shown on Table7. Incremental matrix can be solved with classic MADM:

Table 7 the new Incremental matrix of MADM

	x_1^+	x_2^-	x_3^+
A1	0.54	0.55	0.6
A2	0.7	0.33	0.4
A3	0.46	0.77	0.7
A^{12}	0.562	0.49	0.643
A^{22}	0.787	0.383	0.431

With applying TOPSIS to this matrix, we achieve following permutation:

TOPSIS ➔ $A^{22} > A_2 > A^{12} > A_1 > A_3$

We can continue this process, starting with above matrix and new stepwise coefficients, and apply directly classic MADM technique to this matrix.

10. Conclusions

In the Environment with high degree of complexity, DM is not able to distinguish all alternatives of the system. Therefore its MADM matrix possesses incomplete structure, and decision making in the bases of classic MADM on the initial matrix lead to a non-optimal decision making. What has been shown in this study is how to utilize the Decision Maker Agent which autonomously generates system unspecific alternatives and could be independent of DM. Then, agent adds new generated alternatives to initial matrix. Hence MADM matrix contains more alternatives of system for decision maker. For generating the alternatives which DM was not able to recognize them, from the beginning, the autonomous agent moves step by step on decision surface and uses the path that has been referred in the corresponding algorithm. All new generated alternatives could not be added to initial matrix, because agent generates them autonomously, out of DM control. For this, agent has layers in which some alternatives that could not satisfy DM attitude, are eliminate (Feasibility Analyzer & Dynamic Filter) .These autonomous Decision Maker Agents interact with DM which results to increment the matrix that would be reconsidered and DM, on the bases of classic MADM, could make decision from incremental matrix.

References

[1] Hwang, C.L., Yoon, K., 1981. *Multiple attribute decision making*. Springer-Verlag.
[2] Triantaphyllou, E., Shu, B., Nieto Sanchez, S., Ray, T. 1998. Multi-Criteria Decision Making: An Operations Research Approach. *Encyclopedia of Electrical and Electronics Engineering*, John Wiley & Sons, NewYork 15 175-186.
[3] Howard, R. A. 1971. *Dynamic Probabilistic Systems: Semi-Markov and Decision Processes*, vol. 2 of *Series in Decision and Control*. John Wiley & Sons.
[4] Kaebling, L.P., Littman, M.L., Cassandra, A.R. 1998. Planning and acting in partial observable stochastic domains. *Artificial Intelligence* 101(1–2) 99–134.
[5] Leong, T.Y. 1994.*An Integrated Approach to Dynamic Decision Making under Uncertainty*. Laboratory for computer science MIT/LCS/TR631.
[6] Monahan, G.E. 2000. *Management Decision Making*, Cambridge University press.
[7] Kendall, 2002, *Systems Analysis and Design*. Fifth Edition. Prentice Hall Inc.
[8] Alpert, M.I. 1971.Identification of determinant attributes: A comparison of methods. *Marketing Research*
8(5) 184-191.

[9] Armacost, R.L., Hosseini, J.C. 1994. Identification of determinant attributes using the analytic hierarchy process. *Journal of the Academy of Marketing Science* 22(4) 383-392.
[10] Youngpil, C., Seongbong C., Mooyoung, .J. 2003. Satisfaction assessment of multi-objective scheduleu sing neural fuzzy methodology, *International Journal of Production Research* 41(8) ۱۸۳۱-1849.
[11] Boutilier, C., Dean, T. and Hanks, S. 1999. Decision-theoretic planning: structural assumptions and computational leverage. *Journal of Artificial Intelligence Research* 11 1-94.
[12] Leger, C. 1999. *Automated Synthesis and Optimization of Robot Configurations an Evolutionary Approach*. PhD thesis The Robotics Institute Carnegie Mellon University Pittsburgh, Pennsylvania.
[13] Baroni, P., Guida, G., Mussi, S., Vetturi,A. 1995. A distributed architecture for control of autonomous mobile robots. *In ICAR'95 (*September) 869-877.
[14] Littman, M.L., 1996, "*Algorithms for Sequential Decision Making"*, PhD thesis, Department of Computer Science, Brown University.
[15] Cardon, A., Galinho, A. and Vacher, J.P. 2000. Genetic algorithms using multi-objectives in a multi-agent system. *Robotics and Autonomous Systems* **33** 179-190.
[16] Barraquand, J., Langlois, B., Latombe, J.C. 1992. Numerical potential field techniques for robot path planning. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC **22(2)** (March/April) 224–241.
[17] Menczer, F., Street, W., Degeratu, M. 2001. Evolving Heterogeneous Neural Agents by Local Selection. Advances in the Evolutionary Synthesis of Intelligent Agents, MIT Press 337–366.
[18] Ferber, J. 1999. *Multi-agent systems: an introduction to distributed artificial intelligence*. Addison-Wesley.
[19] Alami, R., Chatila, R., Fleury, S., Ghallab, M., Ingrand, F. 1998 .An architecture for autonomy. *Journal of Robotic Research* **17(4)** (April) 315-337.

Acknowledgements

The authors express their gratitude to A. Hadighi, H.Ghoreishy and M.Shirazi for going through the manuscript.