Metaphor of Politics:
A Mechanism of Coalition Formation

R. Sorbello and **A. Chella**
Dipartimento di Ingegneria Informatica
Universita' di Palermo

R.C. Arkin
Mobile Robot Lab.
Georgia Institute of Technology

Abstract

Hybrid Multi-Agent Architectures support mobile robots colonies moving in dynamic, unpredictable and time varying environments to achieve collective team-oriented behaviors for solving complicate and difficult tasks. The development of a new coalition formation and coordination framework for robot colonies in dangerous, unknown and dynamic environment is outlined. The name of this new framework is Metaphor of Politics (MP), and it loosely takes inspiration from the political organizations of democratic governments. The main characteristic of the proposed framework lies in its dynamic reconfigurability in order to adapt the robot colony to environmental changes.

Introduction

The problem for the coalition formation and coordination of a robots team for complex tasks in dynamic, not predictable environments has been studied in the robotic literature by many researchers (Arkin and Balch, 1998) (Balch and Parker, 2002) (Gerkey and Mataric, 2002) (Fredslund and Mataric, 2002) (Mataric, 1995) (Murphy et al., 2002) (Parker , 1998). In this paper a new hybrid and dynamic framework is proposed. This framework takes inspiration from the political organizations of the democratic governments. The main idea behind the framework is that the leadership is not owned by a single robot, but by a government of robots. The "robot citizens" then execute the tasks according to the government rules. In this way a compromise may be reached among the centralized and the distributed approaches. The goal of the framework is to have a distribution of the planning actions, where each robot saves a deliberative independence status without losing its own reactivity. The agents receive high level goals by the government members and exploits their own deliberative capabilities to choose the faster strategy. The idea is to coordinate a colony of robots that are able to exhibit complex behaviors in order to accomplish a high-level mission but at the same time the framework includes a mechanism to form a new coalition caused by a failure of the government's strategy or a general inefficiency of the whole colony during the reaching of the mission's goals.

Mathematical Model

The MP framework considers a colony composed of H robots and M political parties, with $M \leq H$ to guarantee the presence of at least one robot for every party. Within this framework a set of political issues is associated with every robot, which may express, for example, the individual's attitude towards risk, its dependence on reactive or deliberative behavior, its exploration proclivities, or interest in object recovery. The robots' attitudes towards these issues are represented over the range [-1,1], where 0 means don't care, -1 absolutely not, and 1 absolutely yes. Each party is represented by an ideal prototypical robot, standing for the *central* positions with respect to the political issues that characterize the party. Each robot is identified by N features; for each robot i and party j there is a vector of n issues: $I_i^R, I_j^P \in M^{(n\times1)}$ where $i = 1 \ldots H$, $j = 1 \ldots M$, P = party and R = robot. As an example to describe our model, we consider 3 issues which are identified with the following terms and meanings:

- WELFARE: *Energy of the robot*
- DEFENSE: *Attitude towards risk*
- LABOR: *Amount of work*

Every issue is weighted by a non-negative coefficient (from 0 to + ∞), where the coefficient represents the intensity or the strength of the issue. Every robot R_i and party P_j is represented by a vector with n components:

$$R_i = S_i^R \cdot I_i^R \ , \ \ P_j = S_j^P \cdot I_j^P \tag{1}$$

where S_i^R, S_j^P are diagonal $n \times n$ matrixes containing the weights of the robots' attitudes towards the issues and of the parties issues respectively; R_i and P_j are representative of a robot and of a party in a multi-dimensional space called ROBOT ISSUES SPACE.

There exist several designated political roles within the coalition that a robot can occupy when elected. The heterogeneity of the robots inside the colony is described by a

Roles Matrix (RM) which shows the capability for a robot to qualify for a role in the government; in the case of having H robots and 4 roles the matrix will be of rank $H \times 4$.

Voting Process

The voting process consists of two steps. The first step is *Cluster Identification*, where classical clustering techniques can be applied to our problem for the identification of the membership's groups. The literature provides several candidates; in particular we focus on *Voronoi tessellation*. In the MP framework, the cluster identification step groups the robots of the colony on the basis of their party membership; this choice is based on the consideration that every robot maintains a political orientation depending on the closest aligned political party according to the issues. A robot R_i is deemed to belong to the P_j party if the following condition holds:

$$R_i \in P_j \Leftrightarrow d_{i,j} = \min_k \{d_{i,k}\} \quad k = 1, 2, \ldots M \tag{2}$$

where $d_{i,k}$ is the euclidean distance between the robot R_i and the idealized party position P_k in the ROBOT ISSUES SPACE (figure 1 shows the result of the clustering process). The second step is *Vote Extraction*: the vote expressed by

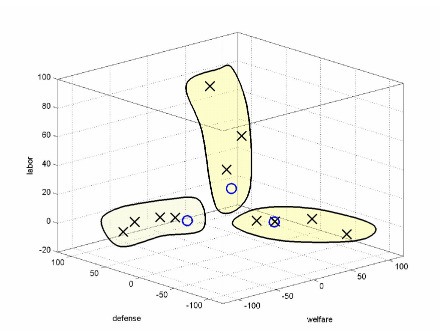

Figure 1: Clustering formation

a robot is simulated by a random number generated in the interval [0,1]. This interval is divided into M sub-intervals, each associated to one of the M parties. The robot's generated value forms is considered as the vote expression a particular party. For every robot i of the colony, the size of each party's subdivision of the interval [0,1] is based on the relative distances of the robot from each party j:

$$d_{i,j}^{REL} = \frac{\sum_{k \neq j} d_{i,k}}{\sum_k d_{i,k}} \% \qquad k = 1, 2, \ldots M; \tag{3}$$

To obtain $\sum d_{i,j}^{REL} = 100\%$ for each robot i, we normalize by a value equal to $1/(M-1)$.

Coalition Formation

The formation of a political coalition which constitutes the new government is made with the support of a linear space, the POLITICAL IDEOLOGY SPACE. This space represents all the robots and the parties belonging to the ROBOT ISSUES SPACE. The mapping between these two spaces is performed by a mapping function $f(\cdot)$ operating between the two representation spaces that groups robots exhibiting similar voting tendencies as evidenced during the voting process. The mapping is based on the following considerations: analogous to an actual parliament, the coalition axis is divided into 3 sections, each representing the ideologies of the left, center and right political viewpoints. The origin point of this linear space is centered to coincide with the pure political center, resulting in negative values being associated with parties aligned to the left and positive values to those towards the right. Two functions, $f_R(\cdot)$ and $f_L(\cdot)$, are introduced in order to determine, when applied to a P_j party, those aspects which respectively characterize trends to the left or right. Each evaluated party will have an overall political trend determined as a compromise between its constituent right and left views on the issues. We represent the overall position of the party P_j with p_j in the coalition (ideology) space using the following heuristic equation:

$$p_j = f(P_j) = f_R(P_j) - f_L(P_j) \ . \tag{4}$$

According to the previous equation, a positive value identifies a rightist party while a negative one identifies a leftist one; values closer to zero identify right-center or left-center parties. The functions are based on suitable vectors of coefficients which weigh the members of the party. The coefficient values are related to the opinions associated with the issues of the parties in order to identify their ideology in the ROBOT ISSUES SPACE. For a linear f, the functional mapping is:

$$p_j = M_R^T \cdot P_j - M_L^T \cdot P_j = (M_R - M_L)^T \cdot P_j \tag{5}$$

where $M_R, M_L \in M^{(n \times 1)}$. A political mass $m_{i,j}$ is associated with each robot i and represents its weight within the voted party j; the calculation of this mass is based on the following function:

$$m_{i,j} = \frac{\sum_{k \neq i} d_{k,j}}{\sum_k d_{k,j}} \qquad \text{if } i \text{ voted for } j\text{, 0 otherwise} \tag{6}$$

where the index k includes all the robots of the colony which expressed a vote for the j party. Every party represented in the POLITICAL IDEOLOGY SPACE is characterized by a mass center dependent on the robots which expressed a vote for the party, and the political masses associated with the robots themselves. This center of mass is obtained using an analogous concept from classical physics:

$$r_{CM}^{(j)} = \frac{\sum_i m_{i,j} \cdot r_i}{\sum_i m_{i,j}} \tag{7}$$

where the index i describes all the robots of the colony which voted j. Figure 2 shows a graphic representation of this situation.

When the mapping process is finished, the coalition which will constitute the new government is formed by adding adjacent parties to the winning party until more than 50% of the total votes cast are reached. The concept of adjacent party is related to the distance between the center of mass of the winning party and the center of mass of the remaining ones. This coalition formation is represented in figure 2 for a case involving 3 parties and 11 robots.

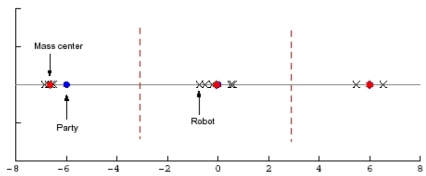

Figure 2: Mass center and coalitions

Role Determination

In our simulation, we choose the following government roles: *Prime Minister* (PM), *Minister of Defence* (MD), *Minister of Communications* (MC), assigned on the basis of the following rules: the PM is chosen from the robots belonging to the winning *party*, while the MD and the MC are chosen between the robots belonging to the winning *coalition* which have not assumed a previous governative role. Representing respectively with $r_k^{(PM)}$, $r_k^{(MD)}$ and $r_k^{(MC)}$ the positions of the robots which satisfy these conditions, and with r_{CM} the position of the center of mass of the winning party, the PM role is assigned to the robot i closer to the r_{CM} center of mass, the MD role is assigned to the robot i positioned to the rightmost extremity of the coalition, and the MC role is assigned to the robot j positioned to the leftmost extremity of the coalition:

$$R_i = PM \quad \Leftrightarrow \quad r_i^{(PM)} = \min_k |r_k^{(PM)} - r_{CM}| \tag{8}$$

$$R_i = MD \quad \Leftrightarrow \quad r_i^{(MD)} = \max_k (r_k^{(MD)} - r_{CM}) \tag{9}$$

$$R_j = MC \quad \Leftrightarrow \quad r_j^{(MC)} = \min_k (r_k^{(MC)} - r_{CM}) \tag{10}$$

A graphic representation of the roles assignment appears in figure 3 and in figure 4.

Conducting Business

The robots comprising the new government produce behavior to achieve their common goals in agreement with the underlying political ideologies of the their coalition. The political ideologies are represented by a *strategy* that the robots must adopt. In the MP framework, two fundamental strategies are used: a leftist progressive (typically reactive) and a right-wing conservative (typically deliberative). In general, the government coalition is constituted by several parties for which these two strategies are the extremes of an overall methodology which changes its characteristics

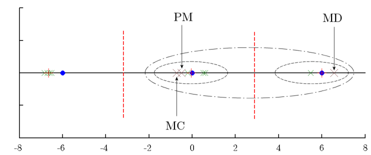

Figure 3: Determine roles in the POLITICAL IDEOLOGY SPACE

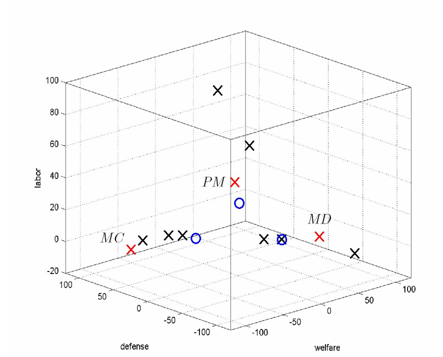

Figure 4: Determine roles in the ROBOT ISSUES SPACE

depending of the formation of the government. A right-center or left-center government will favor either a progressive or conservative strategy based on the weight given to the right or the left components during the formation of the government coalition. A strategy is characterized by a set of parameters which identify various aspects of the robot's behavior; each parameter has values along a continuous interval whose extremes (lower and upper) are associated with the left and right strategies. For every parameter s, for each of M parties there is an associated value so that $s_1 \leq s_2 \leq \ldots \leq s_j \leq \ldots \leq s_M$ where j represents the generic party and s_1, s_M identifies the two extreme parties. The winning coalition is constituted from M' parties with $M' \leq M$. The parameter s_c is only affected by the parties that form the coalition, where acts on the basis of its relative weight within in the coalition. Its value is calculated as a weighted average of the parameters of the coalition parties:

$$s_c = \sum_k a_k \cdot s_k \tag{11}$$

where k refers to the parties which form the coalition. The a_k weight associated with the *k-th* party is obtained by taking into account the V_k votes which it received with respect to the total votes of the coalition:

$$a_k = \frac{V_k}{\sum_h V_h} \tag{12}$$

Mini-Crisis. A mini-crisis is a mechanism which allows the partial replacement of the government with new robots belonging to the existing coalition and business is conducted using the same strategy. This mechanism eliminates the need for re-election of a new government and works to solve inefficiencies like the death, damage, or excessive loss of energy of any current government members, which would negatively affect the behavior of the entire colony. A robot fault/failure requires a change in the Roles Matrix; for instance if the elected robot *i-th* cannot cover its role *k-th* any longer, then the matrix element (k, i) is replaced with a zero value. A mini-crisis is generated when an operating parameter exceeds its limits as well, for instance when the Welfare (which could provide information about the robots' available energy), falls below a critical threshold. A mini-crisis is not as critical as a full re-election that would stress the existing coalition, rejecting the current strategy being used, even if the overall colony behavior was acceptable.
Re-election. The re-election mechanism allows the colony to either reconfirm the previous coalition or change it completely. A re-election is normally caused by the expiration of a fixed time assigned for the government to complete the entire mission (TIME OUT), or in exceptional circumstances for evident deficiencies in the performance of the colony (NO CONFIDENCE), that is not imputable to a single robot but rather to the governing strategy. For instance if mini-crises occur frequently then there is something wrong in the adopted policy and a re-election needs to be conducted.

Experimental Results

Unlike robotic architectures which employ complex central management of a simple-robot colony, the MP framework focuses on coordination of robots which each possess a high degree of autonomy. Every robot is able to express a vote to elect a party, in order to form a political coalition and to identify a governing strategy that permits the accomplishment of mission goals. The MP framework has been implemented using the *MissionLab* simulation environment developed at the Georgia Institute of Technology, including the behavioral states for:

1) ELECTION
2) DETERMINING ROLES
3) CONDUCTING BUSINESS

The interaction among the states of the framework is shown in figure 5. The core of the implemented framework is the ELECTION state which represents a macro-state containing the two following sub-states:

a) VOTING PROCESS
b) COALITION FORMATION

which are responsible respectively for the voting mechanism for all robots and the formation of the political coalitions which constitute new governments. The VOTING PROCESS is constructed with a set of parameters that characterize each robot; these parameters change during the mission's execution relative for to the quality of assigned task execution. Each robot can be assigned a specific low-level task if the robot's role is as a citizen (e.g., exploration, bomb defusing) or a high-level task if the robot is a government member (planning, communication). The COALITION FORMATION phase determines the winning party on the basis of the votes expressed by the single robots. In order to constitute a winning coalition, parties are selected that are closely aligned to the winning one. The ELECTION state determines the winning political coalition which will constitute the new government. Once the elections have completed and the coalition has been formed, the DETERMINING ROLES state determines which robots will cover the governmental roles. Specifically PM, MD and MC are chosen in order to span the entire coalition ideology, thus guaranteeing the government's future strength. The CONDUCTING BUSINESS state allows a winning government to complete the various tasks assigned it, following a strategy which reflects the political trend of the coalition. The government's strategy is a compromise between the strategies of the single parties which compose the coalition. In the finite state automata (FSA) shown in figure 5, the triggers (links) produce the transitions between various states during mission execution. Their meaning is as follows:

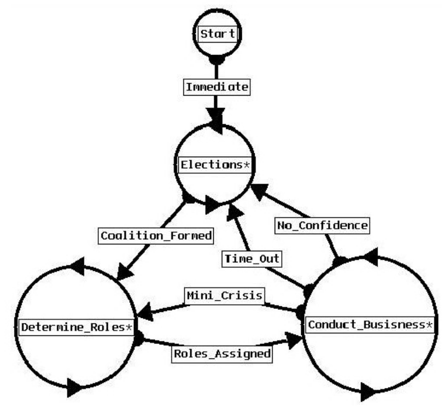

Figure 5: The Robot State Automata Diagram

1. COALITION FORMED: Initiates robot role determination when election and coalition formation processes are concluded.
2. ROLES ASSIGNED: Carries out the strategy embodied by the political trend of the government, occurring when the role assignment phase has finished.
3. MINI CRISIS: Makes a new assignment of government roles, while keeping the political trend of the government unchanged.
4. NO CONFIDENCE: Undertakes a new election as a result of general inefficiency in the whole robot colony.

5. TIME OUT: Generates a new election process when the expiration of time assigned to the government to complete the whole mission occurs.

Figure 5 depicts the MP framework design for use in an unstructured, dynamic, and time-varying environment with unknown or moving obstacles. This domain represents potential use in an application such as mine-defusing by a robotic colony with the presence of high risk for damage to the robots. In order to show MP performances, we created two alternative architectures, named Dictatorship and Anarchy. These alternatives had a twofold purpose: they are at the same time opponents and part of the MP framework. We used in fact them either individually to evaluate their own performance, or inside the MP architecture. In other words, we considered them the two extreme strategies (left and right) between whom our model can dynamically chose, as stated earlier.

Opponent Architectures

These other approaches are distinguished from MP because of the total absence of elections, and thus the impossibility of changing their work in accordance with external conditions and previous results. They are now described in detail:

1. **Anarchy**: This architecture is characterized by the extreme uncertainty of its work. In fact, the robots belonging to the colony explore the area without any fixed strategy, until one of them finds a bomb. When it happens, the robot who found it, calls the nearest one to complete the requested operation. When this activity is completed (either by a successful defuse or with an explosion), all the robots restart their operations as before. Therefore, all the robots are equal in this approach, since each one pursues the same work, without any distinction in terms of roles.

2. **Dictatorship**: This second approach is instead characterized by a strong static distinction in terms of roles. In fact, the entire mission is coordinated by a single robot, called master, that is responsible for the choice of the supporter robot and the communication among the colony. It serves a crucial point for the mission, since its failure cannot be tolerated and would result in the failure of the entire mission. For this reason we tested this approach with three different values of fault probability for the master: 0%, 15% and 30%. Another important role is held by the supporters, the robots which are responsible for helping a robot who finds a mine. When it happens, the finder robot calls the master, who chooses the nearest supporter between the two supporters that are available. The last role is the searcher, that is a robot responsible for finding the mines scattered in the map.

Scenerio Descriptions

The tests focused on a mission in which robots must discover mines that are scattered in the maps, and then they must defuse them. Each bomb has a probability to explode by accident, and a probability to activate its timer; so the defuse operation may not be successful. Moreover some enemy robots (terrorists), with each one being able to release a certain number of mines, can also be present. In particular, four scenarios were developed:

1. 11 robots, no terrorists, with 30 mines placed in the map, time limit of 5 minutes. We took the following parameters: *Dead robots, Defused mines*
2. 11 robots, 2 terrorists that the robots can't kill, 5 mines placed in the map, time limit of 5 minutes. The following metrics were evaluated: *Dead robots, Defused mines*
3. 11 robots, 2 terrorists that the robots can kill, 5 mines placed in the map. The following metrics were used: *Dead robots, Mission Time, Time for Killing Enemies*
4. 11 robots, no terrorists, 25 mines placed in the map. The metrics include: *Dead robots, Mission Time*

Environments

To test the scenarios we used 3 different maps, intentionally developed; in particular they simulate a hangar (see figure 6), a market (see figure 7), and an airport (see figure 8).

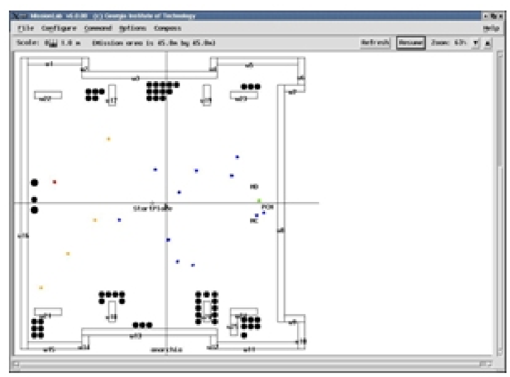

Figure 6: The Hangar Environment

Results

In this section, several graphs are presented, the most significant ones, obtained by testing the various architectures in the different environments, and reporting either the results of each try, or the average achieved from a group of experiments made on a certain scenario and in a particular environment. The graphics illustrated in the next three figures refer

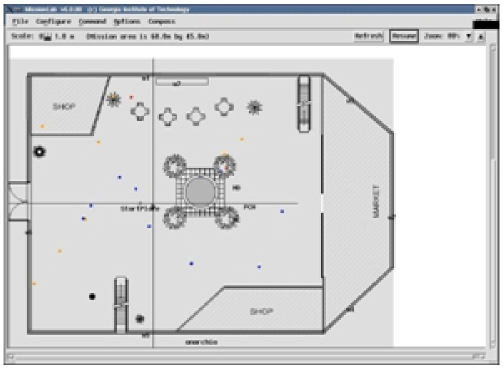

Figure 7: The Market Environment

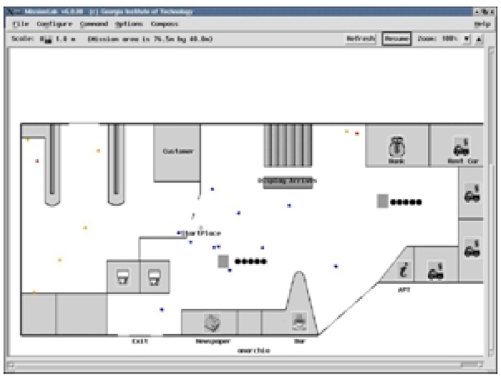

Figure 8: The Airport Environment

respectively to the number of defused mines in the first scenario with a fault probability for Dictatorship's master of 0% (figure 9); the number of robots lost during the mission in the first scenario with a fault probability of 30% (figure 10); and the time spent for mission completion in the fourth scenery with a fault probability of 0% (figure 11), all in the airport map. In these graphs the numbers on the border indicate the number of the experimental group (where each group consists of one hundred experiments), while the numbers on the radii indicate the average value reported for each one. In all these figures it can be seen that the MP architecture adapts its behaviors to the dynamic development of the mission, choosing from time to time the strategy that best adapts to external conditions, thanks above all to a coalition *regeneration* method. This allows the robots to confirm a governing coalition, if mission results are better when compared to previous governments. In fact it can be seen how the line representing MP performance always follows the best point between the opponent architectures, since it chooses the best mix according to their performances. In particular, in figure 9 the exterior line is the best, since it reports the highest values in terms of defused mines: in this case it is the anarchy strategy, but we can see the MP line is tightly close to it, since the architecture reflects what the best strategy is, and doesn't deviate from it often. In figure 10 the best line is

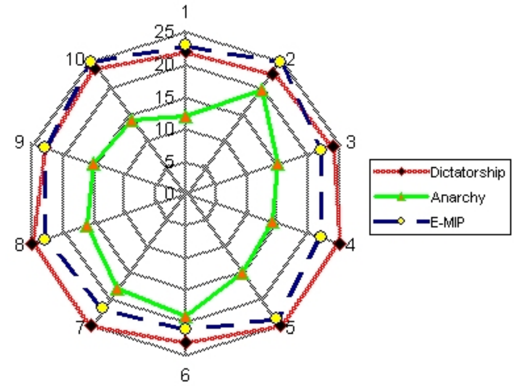

Figure 9: First Scenario : Bombs Defused

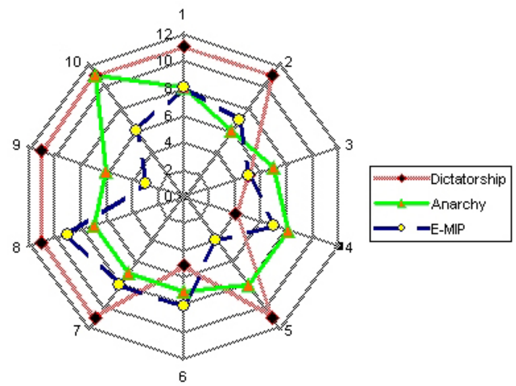

Figure 10: First Scenario : Dead Robots

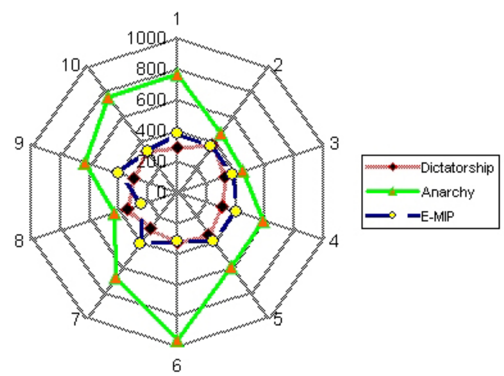

Figure 11: Fourth Scenario : Mission Time

the internal one, because it reports a lower number of dead robots. We see how opponent architectures alternate their performance, crossing themselves several times; MP is instead almost always in the middle, since it selecting among the best one at various times, thus maintaining good performance in all the experimental groups. In the end in figure 11, where the furthest internal line indicates lower mission time, thus better results, MP is very close to the Dictatorship line, because it obtains the highest performance during the experiments.

This phenomenon is even better depicted in the following two pictures, representing the average results, in terms of defused mines, obtained in the second scenario for the different cases of 0% Dictatorship's master fault probability (figure 12) and of 30% Dictatorship's master fault probability (figure 13) after the entire set of experiments. Here is even more evident how the MP architecture is able to choose the best strategy, because in the first case it selects Dictatorship, which has the best performance, but when the master's fault probability is increased thus reducing its result quality, MP robots quickly change the government coalition, moving their position towards the opposite side, thus maintaining enhanced performance, while the Dictatorship performance deteriorates.

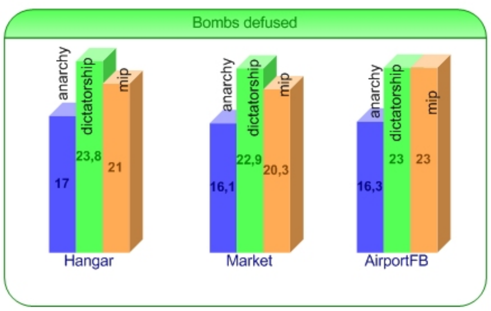

Figure 12: Second Scenario with Dictatorship's Master Fault Probability of 0%

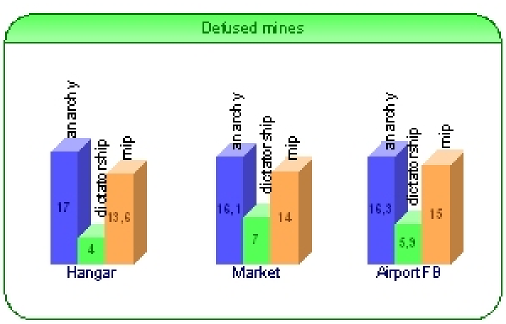

Figure 13: Second Scenario with Dictatorship's Master Fault Probability of 30%

Summary and Conclusions

One of the fundamental and innovative features of the Metaphor of Politics Architecture is its dynamic social structure. It provides a good balance between the cost of forming a coalition and the performance of the coalition itself. It is also capable of forming superior political coalitions for difficult problem solving under conditions of limited information and resources.

References

Arkin, R., and Balch, T.(1998). Behavior-Based Formation Control for MultiRobot Teams. *IEEE Transactions on Robotics and Automation*, vol. 14, no. 6, pp.926–939.

Balch, T., and Parker, L. E.(2002). Robot teams, from diversity to polymorphism. *AK Peters*.

Gerkey, B., and Mataric, M.(2002). Sold! Auction Methods for Multirobot Coordination. *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp.758–768.

Fredslund, J., and Mataric, M.(2002). A General Algorithm for Robot Formations Using Local Sensing and Minimal Communication. *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp.837–846.

Mataric, M.(1995). Issues and Approaches in the Design of Collective Autonomous Agents. *Robotics and Autonomous Systems*, no. 16, pp.321–331.

Murphy, R. R., Lisetti, C. L., Tardiff, R., Irish, L., and Gage, A. (2002). Emotion-Based Control of Cooperating Heterogeneous Mobile Robots. *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp.744–757.

Parker, L. (1998). ALLIANCE: An Architecture for Fault Tolerant MultiRobot Cooperation. *IEEE Transactions on Robotics and Automation*, vol. 14, no. 2, pp.220–240.