Graph Based Representation of Dynamic Planning

Pavlos Moraïtis [1,2] and **Alexis Tsoukiàs**[2]

Abstract. Dynamic planning concerns the planning and execution of actions in a dynamic, real world environment. Its goal is to take into account changes generated by unpredicted events occurred during the execution of actions. In this paper we develop the theoretic model of dynamic planning presented in [12]. This model proposes a graph representation of possible, efficient and best plans of agents acting in a dynamic environment. Agents have preferences among consequences of their possible actions performed to reach a fixed goal. Environmental changes and their consequences are taken into account by several approaches proposed in the so-called "reactive planning" field. The dynamic planning approach we propose, handles in addition changes on agents´ preferences and on their methods to evaluate them; it is modeled as a multi-objective dynamic programming problem.

1 INTRODUCTION

In this paper we develop the formal model of dynamic planning presented in [12] and present several algorithmic aspects. Dynamic planning concerns the planning and execution of actions in a dynamic, real world environment. Its goal is to take into account changes generated by unpredicted events occurred during the execution of actions. According to this approach, changes can come both from a dynamic environment and from the agent himself. Several works are proposed in the so-called "reactive planning" field in order to address planning in a dynamic environment under different approaches (see [4], [5], [6], [7], [15]). Such works propose different techniques in order to react to environmental changes, which may occur during the execution process. We adopt a more general approach since we consider that, in addition, any change may occur in agent's behavior (for any reason, i.e. according to a possible user suggestion) during the execution process, pushing him to change his preferences and consequently his actions or his method to evaluate these preferences. Changes on agent's preferences and on his evaluation methods, are taken into account as revision of three specific structures called *possible plans*, *efficient plans* and *best plans*. To model these structures, we use graphs inspired by the ones described in [13]. Preferences are modeled as criteria in the multi-criteria planning problem we consider. This formalism allows us to present this planning problem as a multi-objective dynamic programming problem. Using dynamic programming in planning problems dates back to Bellman [1], but its use in agency theory has been limited in search algorithms, (see [15]) or in the frame of "universal planning" algorithms (see [14]). Under such a perspective the model we propose allows an agent based on the set of possible actions to achieve a fixed goal, to express his preferences about the benefit he desires to take out (for example, profit, time, pleasure, etc.) by achieving this goal and consequently to define the efficient actions for this end.

Further on by introducing some additional information concerning his preferences, it is possible to define the best plan as the preferred compromise. During the execution of a single action the agent may modify his evaluations (a revision is necessary) or the world may be modified after an unanticipated event (an update is necessary). Such changes (how these are perceived is not considered in this paper) may invalidate the plan under execution in the sense that it could be impossible to follow it or it could be no more convenient. So, the aim of our dynamic planning model is to take into account such changes and to decide what the agent should do.

In the following, section 2 outlines our formalism and describes the multi-criteria planning model. Section 3 presents several algorithmic aspects of the dynamic planning model we propose. We conclude by comparing our research to related work and discussing some problems.

2 THE MULTI-CRITERIA PLANNING MODEL

Let's consider that each agent α_l has to accomplish a set T of tasks in order to accomplish a fixed *goal*. Each task t_i can be decomposed in subtasks necessary to achieve t_i. We can consider that an agent has to go through a set of "states of the world" and more precisely from a state where no task is accomplished (the "nil" state of the world) to a state where all tasks are achieved and therefore his goal is achieved (the "final" state of world). We represent such a situation as an oriented graph. The agent has to execute some actions in order to accomplish his tasks. Each time an action is executed the agent perceives some consequences (for instance a resource is consumed, a distance is computed, a profit is reached etc.). Therefore each time a subtask is achieved the agent is able to register the level of associated consequences on a set of attributes on which he might be able to express his *preferences*.

According to [12] the available information in this planning model consists in:

-a set T of tasks t_i necessary for a fixed goal achievement;
-a set S of possible states of accomplishment s_{li} for each task t_i ;
-a set A of possible actions a_i ;
-a set H of partial orders $\geq_q$ on the set A ($x \geq_q y$: the action x is at least as good as the action y on the partial order $\geq_q$); if some of such partial orders on the set A are at least weak orders, then there exist real values functions h_q, one for each such weak order. We represent with $h_q(a_l)$ the consequences of adopting action a_l under the *private goal* h_q
-a set P of the possible sequences of actions (plans: denoted by ϕ, χ, ψ, etc);

[1] DSS Laboratory, Technical University of Crete
73100 Chania, Crete, Greece, email: moraitis@dias.ergasya.tuc.gr
[2] LAMSADE-CNRS, Université Paris-Dauphine
75775 Paris Cedex 16, France, email:{moraitis, tsoukias}@lamsade.dauphine.fr
*First author, during last academic year, was visiting at University of Cyprus

Finally, we consider that it is possible to define a set G of binary relations $\supseteq_r$ on the set P ($\chi \supseteq_r \psi$: the plan χ is at least as good as the plan ψ on the relation $\supseteq_r$). For the moment, the hypothesis made is that each such binary relation is reflexive ($\forall \chi \in P,\ \chi \supseteq \chi$). This model considers that is possible to establish the relations G on the set *P* from the partial orders *H* on the set *A*.

A concept of "state of the world" is also introduced: a state *w* is a collection of propositions, predicates and/or functions $\langle \Psi, \lambda, \pi_r \rangle$ where: Ψ: is a set of descriptions (under form of propositions) specifying what is true in that state of the world; $\lambda \subseteq T \times S$: is a binary relation associating a task t_i to an accomplishment state s_{li}; π_r: $P \rightarrow R$*:* are functions mapping the set *P* of possible sequences of actions to the reals, representing the binary relations $\supseteq_r$. Of course, such functions exist iff the corresponding relations are at least weak orders (complete and transitive). Often, π_j are computed using the evaluations h_j (a typical case is $\pi_j(p)=\sum_{a_i \in p} h_j(a_i)$).

*An example***:** Let's consider the following situation: an agent α leaves from the city C1 and uses his truck to transport objects to the city C4. The maximum capacity of his truck is two objects. He has two preferences (criteria) about his work, that is maximize his profit and minimize his work time. His possible actions are Go(x, y): go from city x to city y, PutOn(x, Truck): put the object x on the truck, PutOn(xy, Truck): put simultaneously two objects x and y on the truck, Transport(x, y, z): transport object x from city y to city z. We assume that the actions Go(x, y), PutOn(x, Truck), Transport(x, y, z) leave a profit of 1 profit unit, while they generate a loss of 1 time unit. The action PutOn(xy, Truck) leaves a profit of 2 profit units while it generates a loss of 2 time units. Depending on what is his most important preference at a given moment (profit or time), the agent has two possibilities: 1) minimize his work time by loading simultaneously two objects (A, B) in the city C1 (where he leaves) and transporting both objects directly to city C4 2) maximize his profit by going to the city C2 in order to take over an object (D), transporting this object to the city C3 in order to take over a second object (E) and finally transporting both objects to the city C4.

2.1 Possible, Efficient and Best paths

Under such a situation the agent's planning problem can be modeled as a multi-objective dynamic programming problem [12]. Three graphs are established.

Definition 1 *Possible Paths Graph*: a possible paths graph contains a start node corresponding to a nil state (none subgoal is accomplished), an end-node corresponding to a given goal to achieve and a set of intermediate nodes corresponding to intermediate states of the world. Arcs correspond to the set of possible actions an agent can perform to achieve his goal through several subgoals achievement. We denote the possible paths graph as $\Gamma_P = \langle W_P, A_P \rangle$. It should be noticed that Γ_P, operationally, is just a data-base describing the possible states of the world and the arcs among them. It consists in a $W_P \times W_P$ matrix with 0/1 entries, denoting the existence of an action between any two possible states of the world.

Definition 2 *Efficient Paths Graph*: an efficient paths graph represents the set of efficient paths among the possible paths, computed according to the agent's preferences. It represents all "efficient" (not dominated) ways to achieve the agent's goals (Fig 1). Generally it is impossible to find a path that will be the best for all the agent's preferences, (this is an elementary notion in multi-criteria decision aid, see [17]). It is clear however, that there exist paths that are definitely dominated by other ones, in the sense that they are worse under all points of view (all preferences). Let's introduce a dominance relation >>. Given any two possible paths p, p': $p >> p' \Leftrightarrow \forall\ k\ p \supseteq_k p'$ and $\exists k^*: p \supset_{k^*} p'$. The set of efficient paths D will therefore be the set of paths which are not dominated: $D = \{p: \neg\exists p' \in P: p' >> p\}$. We denote the efficient paths graph as $\Gamma_E = \langle W_E, A_E \rangle$. Clearly, $\Gamma_E \subseteq \Gamma_P$.

Definition 3 *Best Paths Graph*: a best paths graph represents the best compromise solution among the efficient paths according to some further additional information (as for instance, an importance relation among his preferences). We make the hypothesis that the agent has such kind of information and therefore he is able to identify a plan p^* such that $\forall p \in D\ \Delta(p^*, p)$, Δ representing a weak order on the set *D*. Under the hypotheses done in this paper, there exist a lot of procedures to identify the "best" compromise solution among the efficient ones (see [8], [9], [3]). We denote the best paths graph as $\Gamma_B = \langle W_B, A_B \rangle$. Clearly, $\Gamma_B \subseteq \Gamma_E$.

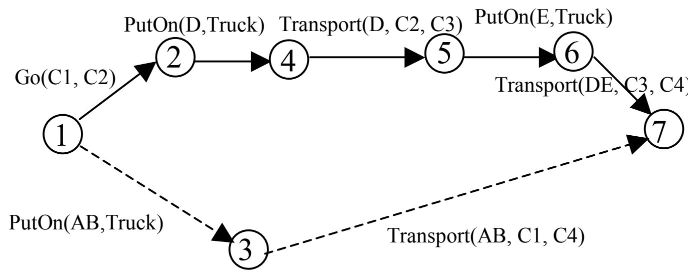

Figure 1: Efficient and best paths of α agent for min-time and max-profit preferences

3 DYNAMIC PLANNING

3.1 Descriptive considerations

During a plan's execution different events may occur such that the agent may modify his evaluations (a revision is necessary) or such that the world is modified (an update is necessary). It is possible that such revisions or updates (hereafter called changes) may invalid the plan under execution. So, what should the agent do? When a change happens, the agent will recognize it and react either following an alternative best plan or constructing a new one. The new plan will have as initial state the state that the agent had reached before the interruption and as final state the same as before (i.e. where the fixed goal is achieved). Depending on the kind of changes, the agent will adopt the most suitable reaction.

3.1.1 Categories of possible changes

We consider the classification of possible changes made in [12]: a) Best paths revision, b) Efficient paths revision, c) Possible paths revision.

1. *Best paths revision* (c_1). A first change that may occur concerns the weak order Δ. For different reasons, the agent may modify the weak order under which the specific best plan has been chosen among the efficient ones. For example, the agent may have modified the priorities or importance of his preferences, (i.e. choose min-time if max-profit has been chosen before, see Fig. 2).

2. *Efficient paths revision* (c_2). A second change that may occur

concerns the states of the world and particularly the functions π^k_j. Actually the way by which the agent evaluates the actions and therefore the plans, as far as his preferences are concerned, can change (for instance the agent may realize that some actions are "more expensive" from what has been considered at the beginning. For example, consider that the action PutOn(AB, Truck) (Fig. 1) under a new estimation generates a loss of 4 time units while it has been before considered that it generates a loss of 2 time units. According to the above assumption, path 1-3-7 where ($\pi_7^p = 3$, $\pi_7^t = 5$) is not any more efficient compared to the path 1-2-4-5-6-7 where ($\pi_7^p = 5$, $\pi_7^t = 5$; considered for min-time and max-profit preferences).

3. *Possible paths revision I* (c_3). A third change that may occur is the elimination of one or more possible actions from the set *A*. The possible consequences of such a change are the following:

(c_{31}) Some states of the world are modified as far as the functions π^k_j are concerned (the sequences under which a state can be reached are now different; the values of some π^k_j can be modified). For example, consider that agent α discovers he is unable to perform the action PutOn(AB, Truck) (Fig. 1) because objects A and B take together more space than the capacity of his truck. Under this situation we also include the case where action(s) eliminated belong to the best plan. That means that the conditions under which some states of the world remain reachable, are modified.

(c_{32}) A state of the world becomes unreachable because all the actions leading to such a state are eliminated. We call such a state as "infeasible state" and we denote it as $w^{\perp}$ (i.e. the state of node 4 (Fig. 1) if agent α is unable to perform the action Transport(D, C2, C3), because the road leading from city C2 to city C3 is blocked).

(c_{33}) A state of the world becomes a "cul-de-sac" in the sense that all actions (arcs) leaving this state (node) are eliminated. We call such a state an "infeasible state" and we denote it as $w^{\perp}$ (i.e. the state of node 4 (Fig. 1) if agent α is unable to perform the action Transport(D, C2, C3)).

Possible paths revision II (c_4). A fourth change that may occur is the availability or necessity of one or more actions, which before were impossible or unforeseen. The possible consequences are the following:

(c_{41}) Some states of the world are modified as far as the functions π^k_j are concerned. A node which was reachable for a certain value of the function π^k_j is now reachable for new values (possibly better). Under such a perspective the new action will connect nodes, which in the original possible paths graph were not adjacent.

(c_{42}) The new actions(s) may create a state of the world, which was not considered in the set *W* (for instance the new action may correspond to the necessity to accomplish a new task or subtask, which was not considered before). For example an unpredicted event (the road is blocked) occurs at the moment when the agent is in the node 4 during the execution of the path 1-2-4-5-6-7 (Fig. 1). The new state of the world not considered in the beginning is (PutOn(D, Truck), blocked(road), see Fig. 3).

3.2 Operational considerations

Different combinations of changes may occur simultaneously, leading to different necessities of re-planning reconsidering the plan under execution. In this paper, we do not care about how a change is perceived by the agent. Two basic problems are of our concern:
1) how to detect a modification and how to classify it according to the previous presentation? 2) how to react to changes in order to adopt a new, possibly better, plan under the new information at hand.

We present here the algorithms corresponding to the second point. Generally, the same dynamic programming approach applies on the modified graphs. The algorithmic aspects of the detection (how the agent detects the type of change occurred) is not detailed in this paper (this is a future work subject). However, we consider that when the execution of the plan is triggered a control program is also executed which may detect one of the following (at least): a) the weak order Δ is modified b) at least one of the relation $\sqsupseteq_k$ is modified c) at least one arc (action) is eliminated d) at least one arc (action) is added.

The agent can found himself in three situations: 1) he is obliged to interrupt the execution 2) he decides to interrupt the execution; this case is not frequent in real world dynamic situations if the environment is the source of changes; however this can be possible in some cases and mainly if the agent himself (or his eventual user) is the source of changes (i.e. the evaluation or the preferences change), 3) he may continue the execution trying to decide a reaction. Let us consider separately the different situations:

Cases 1-2

The agent is in the state w^0. In this case, we consider as p^* the part of the best path not yet executed; in other terms the part of Γ_B going from w^0 to w^f. Moreover, p^{**} is computed as Γ'_B using either Δ' or Γ'_E or Γ'_P (depending on the change occurred), considering as w^{nil} the w^0.

3.2.1 Best Plan Revision

Let us suppose, that agent α is in node 4 having chosen the max-profit and therefore to execute the path 1-2-4-5-6-7 (Fig. 2). In this moment he may decide (for any reason) to change his preference (i.e. min-time) and therefore to transport only one object. In this case he will have to compute p^{**} which correspond to the path 4-7 (Fig 2) by using Δ' on Γ_B. The change concerns the weak order relation, meaning that the criteria (preference/priorities) that were used for the selection of this best plan have changed. All the basic information and the efficient plans are the same, but the previous best plan is no longer the best compromise. The solution is to find the next best compromise from the set of the efficient plans beginning from the point of change and according to the new agent criteria.

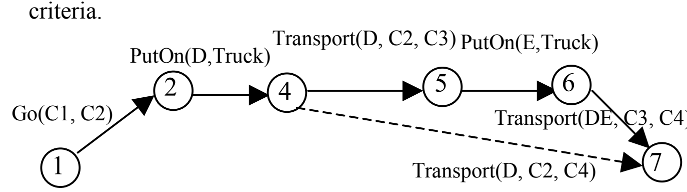

Figure 2. Dotted lines present the new best path for agent α after a preference has changed (min-time)

The following procedure represents the algorithm used to make the best plan revision.

Procedure BEST-PLAN-REVISION (state of the world, preferences)

Build the list of the best plans by using as input the list of the efficient

plans, the state of the world where the problem is detected, and the considered preference(s)
If the list of the best plans is empty then exit
else
    consider the first element of the list of best plans as current best plan
end

3.2.2 Efficient Plans Revision

The change affects an action and more specifically the criterion (i.e. profit or time in our example) parameter of the action. Some actions may be considered in run time as more expensive and the effective or best plan may change. There are two cases:
1) The present best plan is no more efficient and therefore is not any more the best compromise.
2) The present best plan is still efficient, but no more the best compromise.

In both cases since the dominance relation has changed, the algorithm must first rebuild the set of the efficient plans. The next step is to select the best compromise among them. The following procedure represents the algorithm used to make the efficient plan revision.

Procedure EFFICIENT-PLANS-REVISION (modified action, state of the world, preferences)

if modified action belongs to the current best plan then
    if the value of modified action is worse than the value of old action then exit
    else
        -build the list of best plans by using as input the efficient plans, the state of the world where the problem is detected and the considered preference(s)
else
    -build the list of efficient plans by using as input the possible plans and the state of the world where the problem is detected
    -build the list of best plans by using as input the list of efficient plans, the state of the world where the problem is detected, and the considered preferences.
if the list of best plans is empty then exit
else
    consider the first element of the list of best plans as current best plan
end

3.2.3 Possible Plans Revision I

An action or a set of actions from the available action set is deleted. If this action does not belong to the best plan, the execution will continue. If it belongs to the possible plan set and there exists an alternative course that can replace the deleted action, then the agent must rebuild the efficient and best plans. If not, then the agent will examine if it is possible to generate a set of alternative actions that connect the two nodes (pre and post condition states) of the deleted action. The alternative action (or sequence of actions) can be generated by a pool of available actions.

The following procedure represents the algorithm used to make the first type of possible plans revision.

procedure POSSIBLE-PLANS-REVISION I (state of the world, deleted action, preferences)

If the deleted action is not belonging to the current best plan then exit
-Search for alternative action by checking the existence of an alternative action in the set of possible plans
    if not any alternative action exist then
        build an alternative action (or a sequence of actions) by using as input, a start-node (is the node where the problem is detected), an end-node (is the node corresponding to the goal achievement), and the list of the available actions
        if no alternative exists then exit
            else
                -for each possible plan do
                    replace the deleted action by the alternative find
                -for each alternative find do
                    if π(deleted action) is worse than π(alternative) then
                        replace the deleted action in the current best plan by the alternative find
                    else
                        -build the list of efficient plans by using as input the possible plans and the state of the world where the problem is detected
                        -build the list of best plans by using as input the list of efficient plans, the state of the world where the problem is detected, and the considered preference(s)
    else
        -build the list of efficient plans by using as input, the possible plans and the state of the world where the problem is detected
        -build the list of best plans by using as input, the list of efficient plans, the state of the world where the problem is detected, and the considered preference(s)
if the list of best plans is empty then exit
else
    consider the first element of the list of best plans as the current best plan
end

In our example, let´s suppose, like in §3.1 (c_{31}), that the agent can discover that he is unable to perform the action Transport(D, C2, C3) (Fig. 1) because the road is blocked. In this case, the agent will be obliged to go eventually to another city C5 and then follow another road leading from C5 to C3. So a sequence of actions γ_{ij} can be {Transport(D, C2, C5), Transport(D, C5, C3)}.

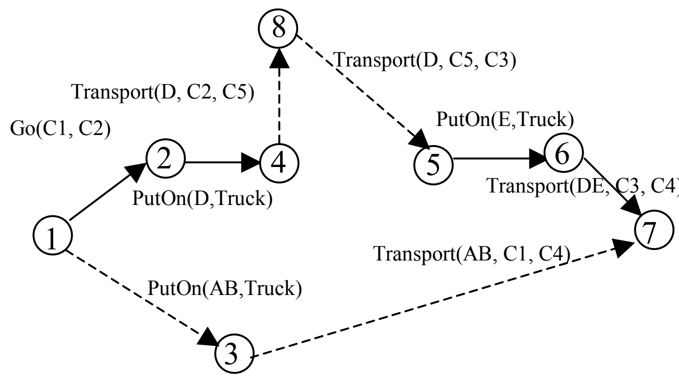

Figure 3. Dotted lines present the new action in Γ'_P graph

3.2.4 Possible Plans Revision II

An action or a set of actions which before were unforeseen or impossible are available or necessary after the modification. Such modifications affect the possible and therefore the efficient and best plans.

-A node that was reachable for a certain value is now reachable for new possibly better values. The new action will connect nodes, which in the original set of plans could not be connected. If the states that are affected (pre and post conditions of the new action) belong to the best plan and they have not yet been reached during the execution, the best plan might no longer be the best. The solution is to construct all plans that include the new action starting from the moment of the interruption, and compare them with the best plan.

-The new action creates a new state of the world. If all actions that leave from this state are the same, then the best plan is not affected. If the actions are different, the agent must rebuild the efficient and best plan sets. It is however necessary to insert all the new actions and states in the possible plans set.

Procedure POSSIBLE-PLANS-REVISION II (state of the world, added action, preferences)

If the added action (or sequence of actions) do not generate a new node then
    for each plan *p* that contains these action (or sequence of actions) do
        if π_p better than π(Current Best Plan) then replace current plan by *p*
else
    if all actions leaving from the new node are the same then exit
    else
        -build the list of efficient plans by using as input the possible plans and the state of the world where the problem is detected
        -build the list of best plans by using as input the list of efficient plans, the state of the world where the problem is detected, and the considered preference(s)
if the list of best plans is empty then exit
else
    consider the first element of the list of best plans as the current best plan
end

Case 3

The agent cannot interrupt the execution of the plan. He perceives the change while being in the state w^0 at time t_0. We have two possibilities:

1.The agent estimates that it is possible to compute a reaction in time t_r inferior to the time necessary to reach w^f or any infeasible state $w^{\perp}$. Considering that the agent is able to compute in which state will find himself after the time t_r (let's denote it w^r) it is possible to apply the reactions presented in the previous section as if the interruption occurred in state w^r. This situation can be possible because we consider that agent has information about his possible plans characteristics, like time execution estimation, rate of success in the past, performed in similar situations. We consider that agent can use such information as criteria for plans choice. In our example we can consider that agent has an estimation of the time he needs to normally reach city C2 by leaving C1.

2. The agent estimates that it is not possible to compute a reaction before the state w^f is reached. With the exception of the cases c_{32}, c_{33} and c_{42} (in the particular situation where the new state is mandatory), the agent will execute the intended plan (although it may be no more the best). In cases c_{32} and c_{33}, the agent will necessarily interrupt the execution when will reaching the infeasible state. He may therefore elaborate an alternative plan starting from any node, which is not infeasible and belongs to Γ_B. In the case c_{42} (and denoting the new mandatory state as w^m) the agent has to verify if an action a_{fm} (from the previous final state w^f to the mandatory w^m) is feasible. If yes, he will just add such an action to the plan, whatever consequences may produce. If not, he has to interrupt the execution to the last state from which it is possible to reach the mandatory state.

4 RELATIVE WORK AND CONCLUSION

As presented in [12], the principal difference of this work compared to others in the field of so-called "reactive planning" [4,5,7,14] (where different techniques are proposed for react to environmental changes), is that we adopt a more general approach. With respect to such papers we additionally consider that any change may occur in agent's behavior (for any reason, i.e. according to a possible user suggestion), pushing him to change his preferences and consequently his actions or his method to evaluate these preferences. Several works (see [10,11]) have been proposed in literature where graph theory and dynamic programming is used for planning purposes. However, such approaches are based on the idea of a "search" on the space of possible states, thus operationally exploring a tree structure resulting from a branching procedure. Our approach is completely different both from a representational point of view (we have a real graph with a single source and sink) and from an algorithmic point of view due to the multi-objective nature of the problem we introduce. For the same reason it is different from planning graphs introduced in [2]. In [16] the idea of rational-based monitoring of plan execution in a dynamic environment is introduced. Their approach is very similar to our classification of possible changes, but limited to environmental ones only. We claim that our model enables a more general characterization of the changes that may occur (i.e. agent preferences and evaluation methods) and how these may affect the computation of a new plan. This work is implemented on JAVA and tested by using several examples. We believe that this paper highlights interesting issues by proposing dynamic planning as a useful mean to reason about changes generated not only by the environment, but also by the agent himself. Our future work will concern the problem of how to detect the changes occurred and how to classify them according to the categories defined in this paper, as well as its integration in the existing model.

References

[1] Bellman R., *Dynamic Programming*, Princeton University Press, Princeton, (1957).

[2] Blum A. L. and Furst M. L., Fast Planning Through Planning Graph Analysis, *Proc. IJCAI-95*, pp. 1637-1642, (1995).

[3] Climaco, J., and Martins, E. 1982. A bicriterion shortest path algorithm, *European Journal of Operational Research*, 11:399-404

[4] Firby R.J., Task networks for controlling continuous processes: issues in reactive planning, *Proc. AIPS-94*, pp. 49-54, (1994).

[5] Gat E., Integrating planning and reacting in a heterogeneous asynchronous architecture for controlling real-world mobile robots, *Proc. AAAI-92,* pp. 802-815, (1992).

[6] Georgeff M.P. and Ingrand F.F., Decision-Making in an embedded reasoning system, *Proc. IJCAI-89*, pp. 972-978, (1989).

[7] Godefroid P. and Kabanza F., An Efficient Reactive Planner for Synthesizing Reactive Plans, *Proc. AAAI-91*, pp. 640-645, (1991).

[8] Hansen P., Bi-criterion path problems, G. Fandel, T. Gal, (eds.), *Multiple Criteria Decision Making: Theory and Applications*, LNEMS **177**, Heidelberg: Springer-Verlag, pp. 109-127, (1980).

[9] Henig M., Efficient Interactive Methods for a Class of Multi-attribute Shortest Path Problems, *Management Science*, **40**, pp. 891 – 897, (1994).

[10] Joslin D. and Roach J., A theoretical analysis of conjunctive-goal problems, *Artif Intell.*, **41**, pp. 97-106, (1990).

[11] Korf R.E., Planning as search: a quantitative approach, *Artif. Intell.*, **33** (1), pp. 65-88, (1987).

[12] Moraïtis, P., and Tsoukiàs, A. Dynamic Planning Model for Agent's Preferences Satisfaction: First Results. In *Intelligent Agent Technology: Systems, Methodologies and Tools, (IAT´99)*, World Scientific Publisher, pp. 182-191, (1999).

[13] Moraïtis P. and Tsoukiàs A., A Multi-criteria Approach for Distributed Planning and Conflict Resolution for Multi-Agent Systems, *Proc.ICMAS-96*, pp. 212-219, (1996).

[14] Schoppers M.J., Universal Plans for Reactive Robots in Unpredictable Environments, Proc. *IJCAI-87*, pp. 1039-1046, (1987).

[15] Stentz A., The Focussed D* Algorithm for Real-Time Re-planning, *Proc. IJCAI-95*, pp. 1652-1659, (1995).

[16] Veloso M.M., Pollack M.E. and Cox M.T., Rational-Based Monitoring for Planning in Dynamic Environment, *Proc. AIPS-98*, (1998).

[17] Vincke P., *Multi-criteria Decision Aid*. New York: John Wiley, (1992).