Abstract Policy Evaluation for Reactive Agents

Krysia Broda and Christopher John Hogger

Department of Computing, Imperial College London
South Kensington Campus, London SW7 2AZ UK
{kb, cjh}@doc.ic.ac.uk

Abstract. This paper describes a method for constructing and evaluating teleo-reactive policies for one or more agents, based upon discounted-reward evaluation of policy-restricted subgraphs of complete situation-graphs. The combinatorial burden that would potentially ensue from state-perception associations can be ameliorated by suitable use of abstractions and empirical simulation results indicate that the method affords a good degree of scalability and predictive power. The paper formally analyses the predictive quality of two different abstractions, one for applications involving several agents and one for applications with large numbers of perceptions. Sufficient conditions for reasonable predictive quality are given.

1 Introduction

Teleo-reactive (TR)-agents were introduced in [16] and further developed in [1] and [18]. Such agents act in response to stimuli received from their environment in such a way to predispose them towards achieving known goals. Their simplest program structure is a set (called a *policy*) of mutually-exclusive production rules of the form *perception* → *action*, usually intended to control durative behaviour: given a current perception the agent performs the corresponding action until acquiring a new perception, whereupon it reacts likewise. We make two key assumptions about TR-agents: they have (i) little or no access to cognitive resources, such as beliefs or reasoning systems, and (ii) only partial observational capability, in that their perceptions may not capture the whole environmental state. A policy identified on this basis is *implicitly* goal-oriented. A significant advantage is the relatively low resources a TR-agent needs for its internal logic; unlike a deliberative agent [13] it does not need computational facilities capable of executing complicated software. Moreover, since the agent's policy is designed to be effective whatever the state in which it finds itself, unexpected exogenous changes in the environment do not cause difficulties. A framework for evaluating policies was proposed in [3] and extended in [4, 5] to use abstraction to deal with scalability, especially in multi-agent contexts. This paper investigates in Theorems 1 and 2 the level of approximation entailed in using abstractions by giving some sufficient conditions for reasonable predictive quality.

Our work is similar to, but different in approach from, those who seek to optimize simple agents, comparable to our own, by the use of Markov Decision

Processes (MDPs) or – when the agents cannot perceive the state's entirety – Partially Observable MDPs (POMDPs) [7, 10, 15]. The key assumption made in these design methods is that beliefs about the agent's current state can be inferred on the basis of its previous action and/or current perception together with beliefs about its previous state, thence enabling a suitable next action to be chosen. This assumption yields algorithms capable of identifying policies that are optimal or near-optimal relative to one's ability to estimate probabilities given the agent's assumed powers of state observation. These methods are very successful when the above key assumption holds, but are more complicated to apply in the multi-agent context where the updating of each agent's beliefs has to consider the combinatorial impact of the other agents' actions upon the state. Our species of TR-agents are also different from those envisaged by [16], where the design of a good policy rests on the assumption that the goal state is totally observable. The content and ordering of the rules constituting the desired policy are inferred by a reductive planning process that constructs and orders rules so that the operation of each one may suitably enable the operation of others, the whole intended to ensure that the goal state eventually becomes achievable. We would also contrast our approach with those methods [9, 12, 14, 19, 21] that rely upon learning. Here the evolving experience of the agent is effectively translated into merit-oriented weightings of the alternative actions available to each perception. The outcome is typically a non-deterministic policy allowing the agent to choose, for its current perception, between alternative actions according to the weightings, which may be interpreted as the relative probabilities of those actions being the best to perform.

The next section describes our framework and presents two abstractions. Subsequent sections detail each kind of abstraction and analyse the approximations imposed on policy evaluation. The paper concludes with a discussion of the ramifications of our results.

2 Overview of Framework

Any world in which our agents operate is capable of assuming various *states*. An agent has three main features: a set $\mathcal{P}$ of *perceptions* it may have of its environment, a set $\mathcal{A}$ of *actions* it may take and a *policy* relating actions to perceptions. We here restrict the language of states, perceptions and actions to be propositional. In any state $o \in \mathcal{O}$, the agent's possible perceptions form some subset $P(o) \subseteq \mathcal{P}$. A *situation* is any pair (o, p) for which $o \in \mathcal{O}$ and $p \in P(o)$. We call the tuple $\langle \mathcal{O}, \mathcal{P}, \mathcal{A} \rangle$ a TR-application. A perception does not, in general, capture the entire world state and the agent normally perceives only limited information about that state. The problem is therefore how to find an optimal policy for a given goal for an agent that (generally) cannot recognize it.

2.1 Situation Graphs

Our framework is based upon a structure called the *unrestricted situation graph* G, which shows the situations that a representative agent called *self* may be in and the possible actions it may take. Each directed arc in G is labelled by some action. When the agent is in a situation (o, p) its possible actions depend only upon p and form a set denoted by $A(p)$. A key feature of our framework is the process of pruning selected arcs from G according to some policy f, to leave the *f-restricted* graph, denoted by G_f. This graph commits the agent to take, in any situation, the action determined by policy f, and shows what will actually happen. We assume that every node in G_f other than a goal situation has a successor, possibly itself. Goal situations are not given a successor, as we are primarily interested in the effectiveness of policies to reach a particular goal and not necessarily in what happens afterwards. These things are summarised in Definition 1 and illustrated in Example 1 using *BlocksWorld*. (Of course, *BlocksWorld* is just an exemplar of a wide range of state transition systems.)

Definition 1. *Let $\langle \mathcal{O}, \mathcal{P}, \mathcal{A} \rangle$ be a TR-application. The* unrestricted situation graph*, denoted by G, is a directed graph whose nodes are all the acceptable situations admitted by the given application. A policy f is a total function from $\mathcal{P}$ to $\mathcal{A}$ and the* restricted situation graph*, denoted by G_f, is the result of pruning all arcs from G except those sanctioned by policy f.*

Example 1 There are 2 blocks on a table and an agent may see either the table ($s0$), or a block ($s1$), or a 2-tower ($s2$) if it exists, and may be holding (h), or not holding (nh), a block. The state is a list of the heights of towers present on the table. (Situations $4a$, $3d$ and $3e$ are possible only if there are several agents and are used in Example 2.) An agent may take one of the actions: `wander` (`w`), `pick` (`k`) or `put` (`t`). See Figure 1. The goal for this example is that at least one agent shall reach state 2 and see the 2-tower (i.e. be in situation $2c$). There is no action prescribed for perception c, since it occurs only in the goal situation. In what follows we will consider the policies Policy 1 and Policy 2, where Policy 1 always takes the `wander` action and Policy 2 is given by $a \rightarrow \mathtt{w}$, $b \rightarrow \mathtt{t}$, $d \rightarrow \mathtt{w}$, $e \rightarrow \mathtt{w}$. Figure 2 shows restricted graphs for these two policies, in which all actions are `wander` except as indicated. The `wander` action is special in that it permits an agent to change its perception without a state change. Depending on the level of abstraction of the model the result of `wander` taken from situation s may, or may not, include s. In this example we assume it cannot be s.

	States
1	[1, 1]
2	[2]

	States
3	[1]
4	[ ]

	p	O(p)	A(p)
a	s0, h	{3, 4}	{`t`,`w`}
b	s1, h	{3}	{`t`,`w`}
c	s2, nh	{2}	{ }

	p	O(p)	A(p)
d	s1, nh	{1, 3}	{`k`,`w`}
e	s0, nh	{1, 2, 3}	{`w`}

Fig. 1. States, Situations and Actions for Example 1

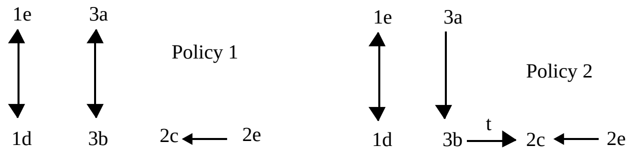

Fig. 2. Policies 1 and 2 (Example 1)

2.2 Measuring Policy Values

We measure the value of a policy f by the method of discounted rewards [10].

Definition 2. *Let f be a policy for a TR-application $\langle \mathcal{O}, \mathcal{P}, \mathcal{A} \rangle$, let $s = (o, p)$ be a situation in G_f and SS be the successor set of s. The* discounted reward *$V(s, f)$, effectively measuring the benefit of the agent proceeding from s, is given by the formula $V(s, f) = \Sigma_{u \in SS}(\chi_{su} \times (\Upsilon_{su} + \gamma \times V(u, f)))$.*

In the above, Υ_{su} is the immediate reward for the action that takes s to u, χ_{su} is the probability that from s the agent proceeds next to u and the factor γ discounts the benefit of taking that action at s. We choose $0 < \gamma < 1$ to reflect the diminishing returns to the agent of performing successive actions. Since we are interested in policies that perform well, on average, from whatever state an agent may find itself, these values of V are used to compute the overall value of f, denoted by $V_{\mathrm{pre}}(f)$, given by the average of $V(s, f)$ taken over all nodes s in G_f. We distinguish two reward values: R for an arc leading immediately to a goal situation and r for all other arcs in G_f. The situations' values are related by a set of linear equations which, since $\gamma < 1$, have unique finite solutions.

There are two issues of scalability for which we propose abstractions. The first occurs when there are several agents. If every combination of agents were to be represented, then each situation would need to include each agent's perception. For applications with up to m perceptions and n agents this could potentially expand the number of situations and policies by a factor of n^m. We choose to approximate the restricted graph by focusing on the actions of a single agent called *self* (see[5]). Ramifications of the behaviour of other agents, necessarily in the same state as *self* but possibly having different perceptions, are reflected in G_f by the use of *exogenous arcs* (denoted by x-arcs). The second issue of scalability arises when the environment's size is increased – for example if there are many blocks. The increase in the number of states is usually accompanied by a gain in the number of perceptions and if every possible perception were to be represented even a small increase in G leads to a large increase in the number of policies. For example, having 10 blocks and allowing a single agent to have the 11 distinct perceptors $s0, \ldots, s10$ would give the agent 21 perceptions in all and one million policies to consider. Results presented in [6] show that both approximations still give reasonable predictions of relative policy values.

3 Formulation for Several Agents

If one were to use our framework to explicitly represent all situations for a group of n agents, the situations would necessarily consist of a state and an n-tuple of perceptions. This is the approach taken in [15], for example. Even for the simple case of *BlocksWorld* with 2 agents and 2 blocks, this gives 17 situations as opposed to 6 situations for a single agent. Nevertheless, we could imagine (but not actually construct) such an unrestricted situation graph; we call it the *group graph* denoted $\mathcal{G}^g$. In [5] for several agents of the same kind (i.e. having the same policy and called *clones*), we introduced the *self graph*, denoted $\mathcal{G}^s_f$, which focuses on a single agent. This graph is a projection of the group graph over the first (or any other) agent for a given policy and we showed it could be used to predict a good *joint policy.* Here, we do not require agents to be clones and instead call the various self graphs *viewpoint graphs*, denoted $\mathcal{G}^v$. It is desirable that the joint policy value should be approximated well by the policy value obtained for any single agent viewpoint. We illustrate for two small examples and in the following section consider under what restrictions the values of policies might be invariant when taken from the viewpoint any individual agent. The notion of TR-application is extended to allow for more than one kind of agent. We use the notation $\langle \mathcal{O}, \mathcal{P}, \mathcal{A}, \mathcal{R} \rangle$, where $\mathcal{R}$ is a set of one or more policies and each agent follows one of them (not necessarily uniquely). We assume here that all agents possess similar perceptive capabilities, although that need not always be so.

Definition 3. *Let $\langle \mathcal{O}, \mathcal{P}, \mathcal{A}, \mathcal{R} \rangle$ be a TR-application with n agents. The list $[(o, p_1), \ldots, (o, p_n)]$ is a* valid group *if in state o it is simultaneously possible for each agent i to have perception p_i. The set $\mathcal{S}_g$ of possible situations is given by $\mathcal{S}_g = \{(o, p_1, \ldots, p_n) | [p_1, \ldots, p_n]$ is a valid group for the n agents $\}$. The set $\mathcal{S}_g$ forms the nodes of the group graph $\mathcal{G}^g$ and its transitions $\mathcal{T}_g$ are derived from the possible transitions any agent could make from each situation: $s = (o, p_1, \ldots, p_n)$ is connected to $s' = (o', p'_1, \ldots, p'_n)$ by action a^i if some agent i in the group can take the action a^i in s to bring about s'. In particular, agent i, when in the individual situation (o, p_i) and taking action a^i, causes itself to make a transition to (o', p'_i) and other agents to their perceptions given by s'.*

That is, each valid group of simultaneous perceptions gives rise to a situation in the group graph and each valid transition of a single agent gives rise to a transition in the group graph. The probabilities on each transition are proportional to those of the individual transitions; *e.g.* if Agent 1 has a non-deterministic action from some situation (o, p_1) with two equi-probable outcomes, then if there are 3 agents these transitions would each have probability 1/6 from any group situation (o, p_1, p_2, p_3). When there are several agents it is possible that *self*'s best policy is to wait for some other agent to change the state, whence it continues. We introduce the `x` action for this purpose. To obtain a viewpoint graph from a group graph, first a particular policy is fixed for each agent and a restricted group graph formed by omitting all arcs except those of the policy given for each agent. Then a projection of the restricted group graph is taken from the point of view of a particular agent i. It is also possible that, from the view of *self*, the state is exogenously changed by another agent. We call this *passive updating* of *self* and label such transitions (in the viewpoint graph) also by `x`.

Definition 4. *Let $\langle \mathcal{O}, \mathcal{P}, \mathcal{A}, \mathcal{R} \rangle$ be a TR-application with n agents and $\mathcal{G}^g$ be a restricted group graph based on the set $\mathcal{S}_g$ of situations of the form $(o, p_1, \ldots, p_n)$ and having set of transitions $\mathcal{T}_g$. Then $\mathcal{G}^v_i$ is the* viewpoint graph *for agent i obtained from $\mathcal{G}^g$ as follows. The situations of $\mathcal{G}^v_i$ are projections of those in $\mathcal{G}^g$ and have either the form (a) (o, p_i), in case $(o, p_1, \ldots, p_n)$ is not a goal situation of $\mathcal{G}^g$, or the form (b) (o, p_j) in case it is, where agent j is responsible for $(o, p_1, \ldots, p_n)$ being a goal situation. The set $\mathcal{T}_v$ of transitions in $\mathcal{G}^v_i$ is given by $\mathcal{T}_v = \{((o, p_i), (o', p'_i))\}$, where $(o, p_1, \ldots, p_i, \ldots, p_n)$ to $(o', p'_1, \ldots, p'_i, \ldots, p'_n)$ is a transition in $\mathcal{G}^g$ and the action for a transition not due to the action of agent i is* `x`*, and otherwise is the action taken by agent i.*

A situation in a viewpoint graph may correspond to several situations in the group graph from which it is derived. The *abstraction function ab*, a mapping from situations in $\mathcal{G}^g$ to situations in $\mathcal{G}^v$, records the correspondences and induces an equivalence relation E_a on the situations in $\mathcal{G}^g$. The E_a equivalence class of a situation s in $\mathcal{G}^g$, denoted $[s]$, is $\{s' | ab(s) = ab(s')\} = ab^{-1}(ab(s))$. In other words, the inverse images of situations in $\mathcal{G}^v$ are the E_a equivalence classes of the situations in $\mathcal{G}^g$. The transition probabilities for $\mathcal{G}^v$ (for Agent i) are derived in proportion to those in $\mathcal{G}^g$ as follows: for a transition between $s1$ and $s2$ in $\mathcal{G}^v$ due to an action a^i of Agent i, the sum of probabilities between each situation in $ab^{-1}(s1)$ and a situation in $ab^{-1}(s2)$ in $\mathcal{G}^g$ due to action a^i of Agent i is computed and divided by $|ab^{-1}(s1)|$. If there are `x`-arcs between $s1$ and $s2$ due to actions of some other Agent j, $j \neq i$ the sum of probabilities over all corresponding arcs between $ab^{-1}(s1)$ and $ab^{-1}(s2)$ is divided by $|ab^{-1}(s1)|$ to give the probability of an `x`-transition between $s1$ and $s2$. The sum of all resulting probabilities of arcs from $s1$ will be 1, since in $\mathcal{G}^g$ the probabilities summed to 1 for each situation in $ab^{-1}(s1)$.

Example 2 *(extends Example 1)* The group graph $\mathcal{G}^g$ is shown in Figure 3, including the various situations, in which the leftmost arrow indicates the status (either seeing the table or a block, and holding (H) or not) of Agent 1 using Policy 1 and the rightmost the status of Agent 2 using Policy 2. All probabilities are 0.5 unless shown otherwise and all actions are `w` except as indicated. There are 17 nodes; nodes 5, 6 and 17 are designated goal situations, when at least one agent is seeing the 2-tower (situation $2c$). The joint policy, obtained using $\mathcal{G}^g$, has the following approximate node values: $v(1) = v(2) = v(3) = v(4) = v(8) = v(9) = v(10) = v(16) = v(11) = -10$; $v(5) = v(6) = v(17) = 0$; $v(7) = v(12) = v(14) = 90$; $v(13) = v(15) = 59$ and total value of 298/14.

To form the viewpoint graphs we use abstraction maps ab_1 and ab_2 between situations in $\mathcal{G}^g$ and $\mathcal{G}^v1$ and $\mathcal{G}^v2$, which are shown together with the viewpoint graphs for the two agents in Figure 4. All probabilities are 0.5 unless indicated otherwise. From the view of Agent 2 there would initially appear to be no possible

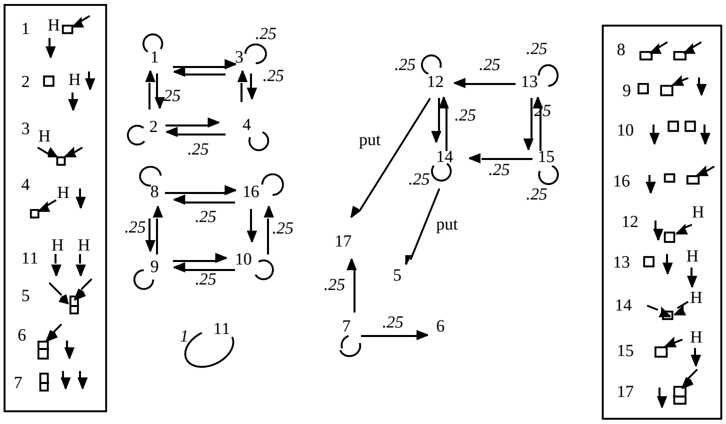

Fig. 3. Group Graph for two Agents using Policies 1 and 2 (Example 2)

exogenous transitions to passively update Agent 2, for since Agent 1 can only `wander` it cannot alter the state. However, $\mathcal{G}^v 2$ has a reflexive `x`-arc from $3b$ to itself arising from the transitions in $\mathcal{G}^g$ between situations 12 and 14 due to the actions of Agent 1. We illustrate the computation for situation $3b$ in $\mathcal{G}^v 2$. The arcs between situations 12 and 14 and the respective arcs on these nodes all arise from the `wander` action of Agent 1 and summing these probabilities in $\mathcal{G}^g$ gives 1. Similarly, the result of summing the probabilities on transitions in $\mathcal{G}^g$ between 12 or 14 and any of the goal situations, corresponding to a transition in $\mathcal{G}^v 2$ between $3b$ and $2c$, is 1. The size of $ab^{-1}(3b)$ is 2, giving probabilities of 0.5 on both arcs from $3b$ in $\mathcal{G}^v 2$. The correspondence between situation 6 and $2c$ for Agent 2 is obtained by case (b) of Definition 4. On the other hand, from the view of Agent 1, there are some obvious exogenous behaviours. When Agent 1 is in situation $3d$ or $3e$, then Agent 2 would necessarily be in $3a$ or $3b$ and, if in $3b$, Agent 2's action would be `put`, so constructing a 2-tower.

If the joint policy is now evaluated from $\mathcal{G}^v 2$, the node values obtained are: $v(3b) = v(2e) = 90$, $v(3a) = 59$, $v(2c) = 0$ and other node values $= -10$. The total value is 189/8, quite close to the value obtained for $\mathcal{G}^g$. However, if a weighted average of the node values is taken, according to the number of elements in $ab^{-1}(s)$, for each s, the average is $(9 \times -10 + 2 \times 90 + 1 \times 90 + 2 \times 59 =$ 298/14. If the joint policy is now evaluated from $\mathcal{G}^v 1$, the node values obtained are: $v(2e) = 90$, $v(3d) = v(3e) = 74.6$, $v(2c) = 0$ and -10 for the remainder. The weighted average is also 298/14, again exactly the value of the joint policy obtained from the group graph. This desirable circumstance does not always prevail, as the next Example shows.

Example 3 This example is from *PlanksWorld*, in which two identical agents aim to dispose of a plank, for which each must be holding a (different) end. This time the joint policy values for the group graph and viewpoint graphs differ.

s in Gg	s in Gv1	s in Gv 2
1	3a	3d
2	3a	3e
3	3b	3d
4	3b	3e
8	1d	1d
10	1e	1e
16	1e	1d
11	4a	4a
5	2c	2c
6	2c	2c
17	2c	2c
7	2e	2e
12	3e	3b
13	3e	3a
14	3d	3b
15	3d	3a

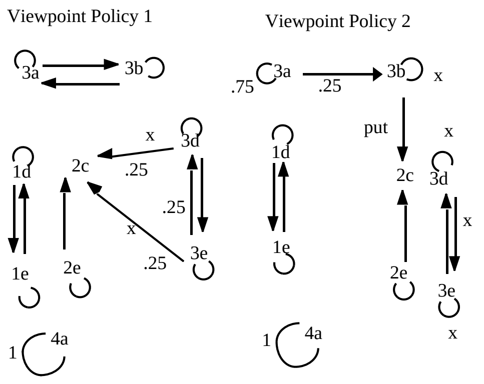

Fig. 4. Viewpoint Graphs $\mathcal{G}^v 1$ and $\mathcal{G}^v 2$ for Example 2

The states and situations are given in Figure 5. Each agent is capable of the

	States
0	[ ]
1	[f]

	States
2	[t]
3	[r]

	p	O(p)	A(p)
a	s0, nh	{0, 1, 2}	{`w`, `x`}
c	su, nh	{1, 2}	{`li`, `w`, `x`}
e	sh, nh	{2}	{`w`, `x`}

	p	O(p)	A(p)
f	sh, h, nr	{2}	{`dr`, `x`}
g	sh, h, r	{ 3}	{ `di`, `dr`, `x`}

Fig. 5. States, Situations and Actions for Example 3

actions `wander`, `drop`, `lift`, `x` and `dispose`. The situation $(0, a)$ is the goal and the states 1-3 are given by describing whether the single plank is (f)lat, (t)ilted or (r)aised. The agents can perceive whether they are holding an end (h) or not (nh), seeing a held or unheld end $(sh$ or $su)$ and, if holding, whether the plank is raised (r) or not raised (nr). It is assumed that an agent can see a held end if it is holding. Policy 3 specifies the following actions for each perception: $a \rightarrow$ `w`, $c \rightarrow$ `li`, $e \rightarrow$ `w`, $f \rightarrow$ `x`, $g \rightarrow$ `di` and the viewpoint graph (projected over Agent 1) and a fragment of the group graph are given in Figure 6, in which all actions are wander unless shown otherwise and unlabelled transition probabilities are 0.5. If the joint policy value is computed from the group graph, the approximate node values obtained are: $v(1a, 1a) = 35$; $v(1c, 1a) = v(1a, 1c) = 42$; $v(1c, 1c) = 55$; $v(0a, 0a) = 0$; $v(3g, 3g) = 100$; $v(2f, 2c) = v(2c, 2f) = 80$; $v(2f, 2a) = v(2f, 2e)) = v(2a, 2f) = v(2e, 2f) = 44$ with total policy value of 610. If instead the joint policy value is computed from the viewpoint graph $\mathcal{G}^v$, the node values obtained are: $v(1a) = 36$, $v(1c) = 50$, $v(2a) = v(2e) = 44$, $v(2f) = 56$, $v(2c) = 80$, $v(3g) = 100$ and $v(0a) = 0$ with approximate total weighted value of 608. (*e.g.* the probabilities of the two arcs incident to situation $2f$ are derived from the single transition to $(3g, 3g)$ (the one to $3g$) and the 9 transitions between situations $(2f, 2c)$, $(2f, 2e)$ and $(2f, 2a)$ (the reflexive arc). In fact, the `wander` arcs contribute $1/3$ and the `x`-arcs contribute 0.5 to the reflexive arc.) Although the two values obtained for the joint policy are very close, they are not equal. In the next section we give criteria which are sufficient to force the two computations to give identical values. These criteria are satisfied for Example 2, but not for Example 3.

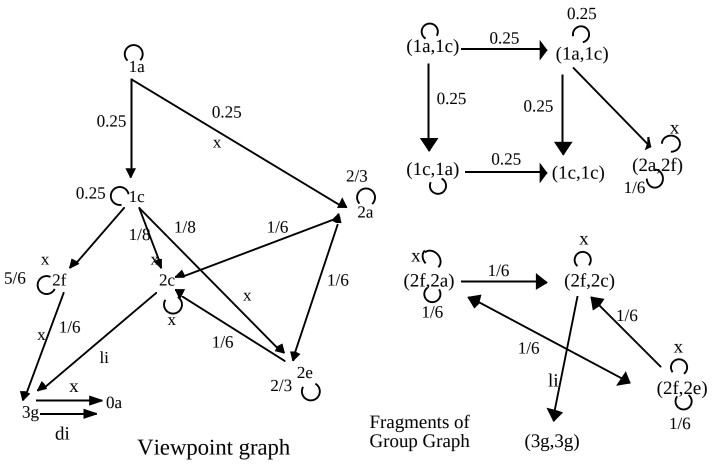

Fig. 6. Graphs for Example 3

4 Relationship between Group and Viewpoint Graphs

Examples 2 and 3 above have shown that the policy values of a group graph and viewpoint graphs derived from it need not be equal. In Example 3, the value of node $2f$ (56) was exactly one-third of the sum of the values of the three nodes in the equivalence class $ab^{-1}(2f)$ (80+44+44). This isn't a coincidence, but does not always obtain; for instance, also in Example 3, the value of node $1c$ (50) is *not* one-half the sum of the values of the two nodes in $ab^{-1}(1c)$ (55+42). Theorem 1 states some sufficient conditions for the above property to hold.

Theorem 1. *Let $\mathcal{G}^g$ be a group graph and $\mathcal{G}^v$ be the viewpoint graph for one of the Agents. Let s be a situation in $\mathcal{G}^v$ and N be the set of situations in $\mathcal{G}^g$ that are mapped to s by ab. Assume also that the rewards on arcs directed to nodes in the same equivalence class of $\mathcal{G}^g$ are equal. Then the quantity $v(s) \times |N|$ is equal to $\Sigma_{n \in N} v(n)$ if either of the following two circumstances holds.*
(i) For each m in $\mathcal{G}^g$ not in N and not of type (ii), and for which there is an arc to m from some node in N, there is an arc from every node in N to every

node in $[m]$, the E_a equivalence class of m, all with the same probability, and either the probabilities on all those kind of arcs are equal for every node in $[m]$ or every node in $[m]$ has equal value.
(ii) For each m in $\mathcal{G}^g$ not in N such that the E_a equivalence class of m is a singleton, exactly one node n in N has an arc leading to m, which is n's only non-reflexive arc, all other arcs from situations in N lead to other nodes in N and the probabilities on these are in certain proportions: for each $n' \in N$, $n' \neq n$ the probability of the reflexive arc must be $pr + p_{n'}$, where pr is the same for all n and the probability of transitions between other nodes in N and n' is $p_{n'}$.

Proof If N is the set of situations in $\mathcal{G}^g$ that are mapped by ab to s in $\mathcal{G}^v$, we shall write $ab(N) = s$. We shall also write $v(s)$ for the value of a situation as computed by the discounted reward formula. The conditions on (i) imply that if any node in N satisfies them, then each node in N satisfies them and except in the trivial case, when N and $[m]$ are both singletons, the two cases are disjoint. Let the size of N be k and each node in N have a bundle of arcs to nodes m belonging to some set DM. We shall denote the value of $ab([m])$ in $\mathcal{G}^v$ by vM and the reward for reaching a node in N or $ab(N)$ by rN and for reaching a node in $[m]$ or $ab([m])$ by rm.
Case (i) Each node in DM is of the type described in (i), hence for each node $m \in DM$ the probability p_{nm} of the arc directed to m from $n \in N$ is the same. The sum of the probabilities of arcs from a node $n \in N$ to other nodes in N is $1 - \Sigma_{m \in DM}(p_{nm})$ and in the viewpoint graph $\mathcal{G}^v$ the probability from $ab(N)$ to itself is $1 - K$, where $K = \Sigma_{m \in DM}(p_{nm})$. There are two sub-cases:
(a) In $\mathcal{G}^v$ the probability of the transition from $ab(N)$ to $ab([m])$ is K_m, where K_m is $\#[m] \times p_{nm}$, where p_{nm} is the same probability for the transition in $\mathcal{G}^g$ from each node in N to each node in $[m]$ and $\#[m]$ is the size of $[m]$. The values of the nodes in $[m]$ may be different, but their mean is Vm.
(b) In $\mathcal{G}^v$ the probability of the transition from $ab(N)$ to $ab([m])$ is $\Sigma_{m' \in [m]} K_{m'} = K_m$, where $K_{m'}$ is the same probability for the transition in $\mathcal{G}^g$ from each node in N to m'. The values of the nodes in $[m]$ are equal and denoted by Vm.
Case (a) Using the discounted reward formula on $ab(N)$ $(= s)$ in $\mathcal{G}^v$ gives $v(s) = \gamma(1-K)v(s) + \gamma\Sigma_{m \in ab(DM)}(K_m.Vm) + (1-K)rN + K.rm$, and hence $v(s) = (\gamma\Sigma_{m \in ab(DM)}(K_m.Vm) + (1-K)rN + K.rm)/1 - \gamma(1-K)) = C/(1-\gamma(1-K))$, where $C = \gamma\Sigma_{m \in ab(DM)}(K_m.Vm) + (1-K)rN + K.rm$.

The values in $\mathcal{G}^g$ of nodes in N can be computed as the sum of the contributions derived from transitions to nodes in N, denoted by $rest_n$, and other transitions (to nodes in DM). The probability from a node in N to each node in $[m]$ is $K_m/\#[m]$. The second contribution is the same for all nodes in N and is given by $\gamma\Sigma_{[m] \subseteq DM}((K_m/\#[m])\Sigma_{m' \in [m]} v(m')) + K.rm + (1-K)rN = C$.
It is required to show that $\Sigma_{n \in N} v(n) = k \times v(s)$. For each $n \in N$,
$rest_n = \gamma\Sigma_{n' \in N}(p_{nn'}.v(n')) = \gamma\Sigma_{n' \in N}(p_{nn'}(C + rest_{n'}))$
$= \gamma C \Sigma_{n' \in N}(p_{nn'}) + \gamma\Sigma_{n' \in N}(p_{nn'}.rest_{n'}) = \gamma C(1-K) + \gamma\Sigma_{n' \in N}(p_{nn'}.rest_{n'})$.
Now, $\Sigma_{n \in N} v(n) = kC + \Sigma_{n \in N} rest_n$, where $\Sigma_{n \in N} rest_n = k\gamma.C(1-K) + \gamma(1-K)\Sigma_{n \in N} rest_n$. This follows since the full expression for $\Sigma_{n \in N} rest_n$ has a unique

solution (and one solution is to set all $rest_n$ equal). Hence $\Sigma_{n \in N} v(n) = kC + (k\gamma C(1-K)/(1-\gamma(1-K)) = kC/(1-\gamma(1-K))$.

Case (b) This time the value of $v(s)$ is $\gamma(1-K)v(s) + (1-K)rN + K.rm + \gamma\Sigma_{m \in ab(DM)}(K_m.Vm)$ and hence $v(s) = C/(1-\gamma(1-K))$.

In the group graph $\mathcal{G}^g$ the probabilities of transitions from nodes $n \in N$ to each node $m' \in [m]$ are the same, denoted by $K_{m'}$, although they may be different for each m'. Then, for a node $n \in N$,

$v(n) = \gamma\Sigma_{[m]\subseteq DM}(\Sigma_{m' \in [m]}(K_{m'}.Vm)) + K.rm + (1-K)rN + rest_n$
$= \gamma\Sigma_{[m]\subseteq DM}(K_m.Vm) + K.rm + (1-K)rN + rest_n$,
where again $rest_n$ is the sum of contributions to $v(n)$ derived from arcs to nodes in N. The first three parts are the same for every n and their sum is C. It is again required to show that $\Sigma_{n \in N} v(n) = k \times vN$ and similar computations for $rest_n$ can be made as before, giving the result.

The proof of **Case (ii)** is simpler and makes similar sorts of calculations. □

Example 2 meets the criteria of Theorem 1 whereas Example 3 does not. For instance, consider $\mathcal{G}^v 1$ of Example 2. For $s = 3a$ and $N = \{1, 2\}$, $[m] = \{3, 4\}$ and case (ia) is satisfied. For $s = 3d$ and $N = \{14, 15\}$, $[m]$ can be either $\{12, 13\}$, and case (ia) is satisfied, or $\{5\}$ and case (ib) is satisfied. On the other hand, in Example 3 criteria (ii) is satisfied for $2f$: $pr = 0.5$ and $p_n = 1/6$ for both $(2f, 2a)$ and $(2f, 2e)$. For $1a$ it is not satisfied, as the reader can easily check.

Although seemingly restrictive, in practice the restrictions on probabilities are often nearly satisfied. Even when not, the proof method shows that, unless the relevant probabilities are wildly variant, the two quantities will still be fairly close since the viewpoint policy averages the various probabilities as if they were equal. This result gives some foundation to our empirical results, obtained in [6], which show that the ranks of the viewpoint policy values, computed using $\mathcal{G}^v$, *are* a good guide to the ranks of the group policy values, computed using $\mathcal{G}^g$.

If the criteria are not met some variation should be expected between the joint policy value as computed by the group and viewpoint graphs. In particular, paths may exist in the viewpoint graph that are not realizable. The viewpoint graph of Example 3 contains the path $\{1a, 1c, 2e, 2c, 3g, 0a\}$, which abstracts the real path $\{(1a, 1c), (1c, 1c), (2e, 2f), (2c, 2f), (3g, 3g), (0a, 0a)\}$, but also implicitly includes impossible paths. For instance, after starting from the possible transition $\{(1a, 1a), (1c, 1a)\}$ Agent 1 cannot move to $2e$. In this example every path in the viewpoint graph corresponds to at least one path in the group graph, but if there are 2 planks and 2 agents, then paths can be found in the viewpoint graph that do *not* correspond to any realizable path. The viewpoint graph has abstracted away details of the groups, and although there may be arcs leading through situations $\{s1, s2, s3\}$ the group that occurs as a result of the first transition may not be a correct one from which to make the second. We call this the *group incoherence problem.* In extreme cases, the valuation of nodes that apparently, but incorrectly, lead to a goal situation can inflate the policy value, so that a bad policy appears better than it really is. On the other hand, the extreme case appears to be fairly rare, so the benefit of viewpoint graphs for scalability outweighs the disadvantages due to the group incoherence problem.

5 Policy Abstraction

There is another way to obtain abstractions. Consider an agent operating in a *BlocksWorld* with many (e.g. 10) blocks; assuming just the actions `wander`, `pick` and `put` there are 21 perceptions and over a million policies to evaluate. The number of policies can be reduced by generalising the perceptions. For example, it could be that an agent can, and only needs to, detect a tower of height $= 0$, < 5 or $= 5$ or > 5. This generalisation can be seen either to be an enhancement of the capabilities of the agent, for example by allowing it to perceive disjunctions, as in seeing a tower of height 1, 2, 3 or 4 (the perception < 5), or by giving it more power to sense the height of a tower; or to be an increase in expressiveness of the policy language, for instance by using first order logic and allowing perceptions of the form $\{size(x), x < 5, x > 0\}$. Either way, not only do the perceptions need to be abstracted, but also several states may need to be combined in order for situations of the form (o, p) to be meaningful. In [4] we investigated this kind of abstraction, and our simulation studies showed it still gave good ranking charts for policies in cases where the number of policies was too large for individual computation. The discrepancies are again due to a coherence problem and to explain it we consider what approximations are involved in calculating policy values for such abstractions.

5.1 Generic Situations

From any TR-application $\langle \mathcal{O}, \mathcal{P}, \mathcal{A} \rangle$, we can form a *generic TR-application*, in which the actions remain unchanged, but the states and perceptions are generalised, which means to introduce, respectively on $\mathcal{P}$ and $\mathcal{O}$, equivalence relations E_p and E_o. The $E_p(E_o)$ equivalence classes are called *generic perceptions(states)* and if perceptions p_1 and p_2 are E_p *equivalent* they will always specify the same action. We could require that an agent is capable of taking the same actions for all perceptions in each E_p equivalence class, or that the policy specifies such an action, but it is not necessary, since if, for some generic perception P, the action specified is not possible for an actual perception in P it could be modelled by a failed action. For each generic state O and generic perception P the *generic situation* $S = O \times P$ is disjoint from all others, which is important since it ensures that no policy can specify two different actions for any real situations.

Example 4 This can be illustrated straightaway for *BlocksWorld* by using a generalisation with just two generic states $[e3]$ and $[ne3]$, denoted by 1 and 2, and three seeing perceptors, $s0$, $s3$ and sx, the latter denoting "seeing neither the surface nor a 3-tower". This yields the 6 perceptions $a - f$ given by $\{(s0, h), (sx, h), (s3, h), (s0, nh), (sx, nh), (s3, nh)\}$. This abstraction suits the goal of building a 3-tower from an arbitrary but sufficient number of blocks. The situations and transitions for the policy whereby the agent, if seeing a tower of height neither 0 nor 3 (sx), can `pick` if not-holding (nh) or `put` if holding (h), but in all other cases wanders, is shown for 3 blocks in Figure 7 (perception c is impossible). The intended goal is the situation $1f$ (i.e. $([e3], (s3, nh))$).

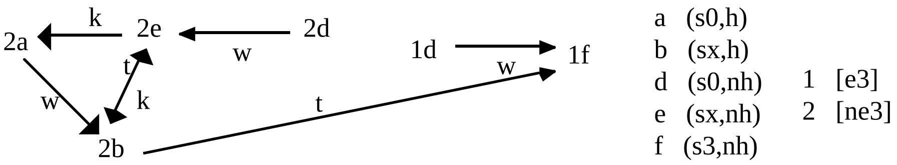

Fig. 7. Using generic situations (Example 4)

A comparison with any standard restricted graph G for 3 or more blocks shows a second incoherence problem called *piecewise incoherence*. In the generic restricted graph there is a path through situations $\{2e,\ 2b,\ 1f\}$. However, this path could never actually occur – both parts of the path from $2e$ to $1f$ are possible, but not in succession. The situation $2e$ corresponds to an agent seeing a tower of height 1, 2 or $\geq$4 and no 3-tower in existence. The policy specifies the `pick` action causing the agent to move to $2b$. In fact, that means the agent could not have been seeing a tower of height 1 or of 4 when in situation $2e$ and the agent must now be seeing a tower of height 1 or $\geq$4, for which the policy specifies a `put` action. The outcome of this action in this particular circumstance could never be $1f$. In G there would be nodes from which a 3-tower cannot be built due to `pick` and `put` actions between situations where the agent sees a tower of height 2 or $\geq$5 and is not holding. For 3 blocks and the above policy G has a value of 51.6, whereas the graph in Figure 7 has a value of 71.5 (where node values are weighted by the number of concrete situations represented by each generic situation and assuming equi-probable transitions in both cases).

The reader may think the problem could be overcome by enhancing the agent with an extra sense, *e.g.* allow it to recall its previous action, so distinguishing between having arrived at $2b$ via $2e$ or via $2a$. This splits the b perception into two, one in which the agent remembers its previous action was `wander`, and one in which it remembers it was `pick`, but it results in non-disjoint generic situations unless a similar perceptive capability obtains in the standard graph G and illustrates the care that must be taken when constructing generic graphs.

5.2 Evaluating Generic Policies

This section discusses the relation between the policy value of a generic graph, and the policy value of the non-abstracted graph for the corresponding policy. The analysis made in the proof of Theorem 2 will also yield a criterion that guarantees no piecewise incoherence in a generic graph.

Definition 5. *Let $\langle \mathcal{O}, \mathcal{P}, \mathcal{A} \rangle$ be a TR-application and E_p and E_o be equivalence relations on the sets $\mathcal{O}$ and $\mathcal{P}$ respectively. Then $\langle E_o, E_p, \mathcal{O}, \mathcal{P}, \mathcal{A} \rangle$ is the* generic TR-application *based on $\langle \mathcal{O}, \mathcal{P}, \mathcal{A} \rangle$ and the set of generic situations is $\{S | S = O \times P\}$, where O and P are, respectively, E_o and E_p equivalence classes. $\langle \mathcal{O}, \mathcal{P}, \mathcal{A} \rangle$ is called the parent application.*

The elements of a generic situation that also exist in its parent application are called *concrete situations*. The probability of a transition from $S1$ to $S2$ in a restricted generic graph G_a is computed as the mean of the probabilities of transitions from a concrete situation in $S1$ to a concrete situation in $S2$ and is denoted by χ^a. We also make two assumptions: (a) the rewards on any transition leading to an element of a generic situation $S1$ are the same, and (b) if generic policy F specifies $P \rightarrow a$, then the corresponding policy f for the parent application specifies the rules $p \rightarrow a$ for every $p \in P$. Assumption (a) imposes the restriction on the equivalence classes E_p and E_o that goal situations and non-goal situations cannot be equivalent.

Theorem 2. *Let G_a be a restricted graph for a generic TR-application and G_f be the restricted graph for the parent application using corresponding policy f. Then, for each generic situation S in G_a, $k \times V_S = \Sigma_{i \in S} v_i$ if the probabilities on each transition between S and U in G_a are the average of the transition probabilities between each $i \in S$ and $j \in U$. (V_S and v_i are the values of situations S and i in G_a and G_f and k is the number of concrete situations in S.)*

Proof For simplicity, zero probabilities are assigned to non-existent transitions. Let S be a generic situation in G_a. Then the sum of values of concrete situations in S is given by $\Sigma_{i \in S} v_i = \gamma \Sigma_{i \in S} (\Sigma_{j \in U, U \in G_a} (\chi_{ij} . v_j)) + \Sigma_{i \in S} (\Sigma_{j \in U, U \in G_a} \chi_{ij} . r_{ij})$, where r_{ij} is the reward on the transition between i and j if it exists (and is irrelevant otherwise). By the assumption (a) each of r_{ij} is the same for and $j \in U$ and all $i \in S$, so the contribution due to reward values can be simplified to $\Sigma_{U \in G_a} (r_{SU} \Sigma_{i \in S, j \in U} \chi_{ij})$.

The quantity $k \times V_S$ is given by $k \gamma \Sigma_{U \in G_a} (\chi^a_{SU} . V_U) + k \Sigma_{U \in G_a} (\chi^a_{SU} . r_{SU})$, The contributions due to reward values are the same in both cases for each U since $k \times \chi^a_{SU} = k(\Sigma_{i \in S, j \in U} \chi_{ij})/k$. The other contribution to $\Sigma_{i \in S} v_i$ can be written as $k \gamma \Sigma_{j \in U \in G_a} (\chi_{Sj} v_j)$, where χ_{Sj} is the mean of the transition probabilities between each concrete situation in S and each concrete situation $j \in U$. If the χ_{Sj} are further averaged over each j, each to be equal to χ_{SU}, then the contribution becomes $k \gamma (\Sigma_{U \in G_a} \chi_{SU} \Sigma_{j \in U} v_j)$. By comparing the expressions for $k \times V_S$ and $\Sigma_{i \in S} v_i$, it can be seen they would be equal if V_S were the average of v_i, $i \in S$. □

In other words, the policy value using generic situations is obtained by assuming that the transition probabilities between generic situations are an average of the actual transition probabilities and that the node values are an average of the concrete situation values. Therefore, in the case that some transitions do not exist, piecewise incoherence is a possibility. In cases where the transition probabilities between the concrete situations making up two generic situations vary widely and/or the values of the concrete situations also vary widely, the generic policy will not be a good reflection of the policy using concrete situations.

In Example 3, notice that if there are many blocks then the probability of situation $2b$ occurring when the agent is seeing a tower of height > 4 is much increased. Thus the probability of the arc between $2b$ and $1c$ would, in practice (i.e. if measured by simulation), be quite small compared with that of the arc between $2b$ and $2e$. This would cause a corresponding reduction in the policy value and improve the approximation, since the generic policy value over-estimates the contribution to the value of $2b$ made by the arc to $1c$. The absolute policy values are not as crucial as their relative ranking – even if the policy values computed using G_a are higher than those computed using the full graph G_f, if the values are ranked in the same order in both cases this will still allow for the best policies to be found. In experiments conducted so far (see [4, 6]) this has been the case. This abstraction has some similarities with that introduced in [11], where a theorem similar to Theorem 2 is quoted. In the circumstances when a transition in G_a between (O, P) and (O', P') implies there is a transition between every $(o, p) \in (O, P)$ and $(o', p') \in (O', P')$ generic policies always give reasonable approximations. Piecewise coherence cannot then occur, since the destination in (O', P') of a concrete transition from (O, P) would always be a source for the next transition from (O', P') to some other generic situation. Abstractions satisfying this criterion were considered in [17] and arise naturally when the goal situation is also changed to reflect the changes in the environment due to scaling; *e.g.* in *BlocksWorld* this kind of goal might be to build a tower of all available blocks, or in *PlanksWorld* it might be to dispose of all planks however many there were initially.

6 Conclusions and Future Work

We have analysed the approximations involved in using abstractions to evaluate policies for TR-agents, in order to test the predictive quality of such abstractions in contexts involving several agents and/or many situations. In the case of several agents we approximated the group behaviour by focusing on a single agent and Theorem 1 shows that the policy values are generally affected and may be subject to group incoherence. This phenomenon is more likely in case there are few states and many perceptions for each state; however, as we assume fairly simple agents, this circumstance appears to be relatively uncommon, which is borne out by our empirical studies in [5, 6]. Moreover, it is less likely in case of a large number of agents, since all perceptions of a given state will be more common, in turn making exogenous transitions in $\mathcal{G}^v$ more likely to occur in practice.

We are investigating the benefits obtainable when agents possessing different perceptive capabilities operate in the same environment. For example, some agents may be endowed with global perceptions, and be capable of few actions, whereas other agents may be capable of more specific actions and perceptions. The former kind of agent could act as an information source for other agents.

We also analysed the approximations due to perceptual abstractions and we found they could give rise to piecewise incoherence and that this was a more common phenomenon than group incoherence. However, empirical results in [4, 6] show that in case the transition probabilities are well estimated, the abstract policy values give fair relative predictions for exact policy values. The results depend on the particular generic situations chosen and this is a topic for our future investigation, together with a comparison of our work with that of [11]. well under changing environmental conditions.

References

1. Benson, S. Inductive Learning of Reactive Action Models, PhD, Dept. of Computer Science, Stanford University, 1996.
2. Benson, S. and Nilsson, N. Reacting, planning and learning in an autonomous agent, Machine Intelligence 14, Eds. Furukawa, K., Michie, D. and Muggleton, S, Clarendon Press, Oxford, 1995.
3. Broda, K. , Hogger, C.J. and Watson, S. *Constructing Teleo-reactive Robot Programs*, Proc. 14th European Conf. on A.I. (ECAI-2000), Berlin, pp. 653-657, 2000.
4. Broda, K. and Hogger, C.J. *Designing and Simulating Individual Teleo-Reactive Agents*, Poster Proceedings, 27th German Conf. on AI, Ulm, pp. 1-15, 2004.
5. Broda, K. and Hogger, C.J. *Policies for Cloned Teleo-Reactive Agents*, 2nd Conf. on Multi-Agent System Technologies, Erfurt, LNAI 3187, pp. 328-340, 2004.
6. Broda, K. and Hogger, C.J. *Determining and Verifying Good Policies for Several Cloned teleo-Reactive Agents*, Intelligent Systems Journal, MATES Special Issue, 2005 (to appear).
7. Cassandra, A.R., Kaelbling, L.P. and Littman, M. Acting Optimally in Partially Observable Stochastic Domains, *Proceedings 12th National Conference on AI (AAAI-94)*, Seattle, pp 183-188, 1994.
8. Chades, I., Scherrer, B. and Charpillet, F. Planning Cooperative Homogeneous Multiagent Systems using Markov Decision Processes, *Proc. of the 5th International Conf. on Enterprise Information Systems (ICEIS 2003)*, pp 426-429, 2003.
9. Dickens, L. Learning through Exploration, MSc Dissertation, Dept of Computing, Imperial College, 2004.
10. Kaelbling, L.P., Littman, M.L. and Cassandra, A.R. Planning and Acting in Partially Observable Stochastic Domains, *Artificial Intelligence* 101, pp 99-134, 1998.
11. Kersting, K. and De Raedt, L. Logical Markov Decision Programs and the Convergence of Logical TD(λ), *Proc. of ILP2004*, LNAI 3194, pp180-197, 2004.
12. Kochenderfer, M. Evolving Hierarchical and Recursive Teleo-reactive Programs through Genetic Programming, *EuroGP 2003*, LNCS 2610, pp 83-92, 2003.
13. Kowalski, R.A. and Sadri, F. *From logic programming to multi-agent systems*, in: Annals of Mathematics and Artificial Intelligence, 25, pp. 391-419, 1999.
14. Mitchell, T., *Reinforcement Learning*, in Machine Learning, McGraw Hill, 1997.
15. Nair R., Tambe, M., Yokoo, M., Pynadath, D. and Marsella, M. Taming Decentralised POMDPs: Towards Efficient Policy Computation for Multiagent Settings, *Proc. of the 18th Int. Joint Conf. on A.I. (IJCAI-03)*, pp 705-711, 2003.
16. Nilsson, N.J. *Teleo-Reactive Programs for Agent Control*, Journal of Artificial Intelligence Research, pp. 139-158, 1994.
17. Nilsson, N.J. *Learning Strategies for Mid-level Robot Control: some preliminary Considerations and Results*, Report, 2000.
18. Nilsson, N.J. *Teleo-Reactive Programs and the Triple-Tower Architecture*, Electronic Transactions on Artificial Intelligence 5:99-110 (2001)
19. Ryan, M.R.K. and Pendrith, M.D. An Architecture for Modularity and Re-Use in Reinforcement Learning, *Proceedings 15th International Conference on Machine Learning*, Madison, Wisconsin, 1998.
20. Snedecor, G.W. and Cochran, W.G. Statistical Methods, Iowa State Univ. Press, 1972.
21. Sutton, R. and Barto, A.G. *Reinforcement Learning An Introduction*, MIT Press, 1998.