Practical assumptions for planning under uncertainty

Juan Carlos Saborío[1] and Joachim Hertzberg[1,2]
[1]*Institute of Computer Science, University of Osnabrück, Wachsbleiche 27, Osnabrück, Germany*
[2]*DFKI Robotics Innovation Center (Osnabrück), Albert-Einstein-Straße 1, Osnabrück, Germany*
{*jcsaborio, joachim.hertzberg*}*@uni-osnabrueck.de*

Keywords: Problem solving, planning under uncertainty, plan-based robot control.

Abstract: The (PO)MDP framework is a standard model in planning and decision-making under uncertainty, but the complexity of its methods makes it impractical for any reasonably large problem. In addition, task-planning demands solutions satisfying efficiency and quality criteria, often unachievable through optimizing methods. We propose an approach to planning that postpones optimality in favor of faster, satisficing behavior, supported by context-sensitive assumptions that allow an agent to reduce the dimensionality of its decision problems. We argue that a practical problem solving agent may sometimes assume full observability and determinism, based on generalizations, domain knowledge and an attentional filter obtained through a formal understanding of "relevance", therefore exploiting the structure of *problems* and not just their representations.

1 INTRODUCTION

Planning in AI is the process of deliberately reasoning about and choosing actions that help an agent achieve its goals. This behavior is goal-directed and its actions may modify the agent's environment. Instead of "lower level" actions related to sensing and motor control, we assume a higher level of abstraction in terms of primitive *commands* (Ghallab et al., 2016). These commands or tasks are the object of deliberation and provide a rich modeling framework to represent and solve practical problems, such as serving a cup of coffee or finding a set of keys. Many if not all of these everyday tasks, however, carry varying degrees of uncertainty. As a normative decision theory, expected utility theory (EUT) solves uncertainty by recommending actions that maximize a combination of utility and probability (cf. (von Neumann and Morgenstern, 1944)). In most practical domains, however, estimating the value of different outcomes and determining optimal decisions is too slow or simply infeasible. The underlying state and action representation for a decision-making agent might also lead to massive search spaces due to combinatorial explosion known as the *curse of dimensionality*. This severely limits planning and decision-making under uncertainty in robots and similar agents.

These tasks are often modeled as a Markov Decision Process (MDP) or more generally as a Partially Observable MDP (POMDP). Advances have been made to speed up planning for (PO)MDP's and there has been relative success when transferring these methods to robot control. Often, however, these advances require arguably small POMDP's, human intervention or do not take advantage of potential, practical simplifications derived from the structure of problems and not just of their formal representations. Dimensionality reduction techniques such as state aggregation or point-based belief estimation are also based on fixed features that may not generalize well across domains, for robots or agents attempting to solve different types of problems.

We argue that practical problem-solving agents must not necessarily assume problems are entirely probabilistic or non-deterministic. Whether from experience or a given model, an agent may assume *practical determinism* for specific subproblems or choices known to *behave in a certain way*. In addition, efficient task planning for robots and other agents requires, at the moment, a satisficing (and not an optimizing) approach. Since much of what determines whether two states or beliefs are similar or not, and whether an action contributes to reaching a goal can be summarized by what we intuitively understand as *relevance*, we borrow this idea and attempt to formalize it in order to produce context- and goal-sensitive grouping criteria. For example, if a robot's goal is to exit a room all state descriptions not limiting the completion of this task (eg. cup on table, cup not on table, etc.) are irrelevant and all variations reduce to the same state, and all actions not leading to the door are also *potentially* irrelevant. Conversely, if the robot must exit the room while holding the cup, then its location becomes relevant. Exceptions can still be discovered through guided exploration.

This paper outlines a research project whose main interest is understanding and formalizing mechanisms for fast and efficient state, action and possibly belief estimation, in order to reduce the dimensionality of otherwise complex decision processes. Its expected contributions are a simplified approach to planning under uncertainty for robots, within the context of practical problem solving. Potential experimental scenarios include task-planning in robots with manipulators, in open ended domains with loosely specified rules. The following sections review previous work and briefly introduce the (PO)MDP framework. Based on the requirements for robot task-planning, we propose an architecture to support satisficing behavior in task planning.

2 RELATED WORK

Advances in planning under uncertainty can be traced back to extensions of classical methods relying on possible world semantics (cf. (Thiébaux and Hertzberg, 1992), (Kushmerick et al., 1994), (Boutilier et al., 1996)), but most current work focuses on efficiently solving large MDP's and POMDP's. Efficient MDP solvers include sparse sampling using a generative model (Kearns et al., 2002) as well as UCT (Kocsis and Szepesvári, 2006). TExplore combined this sample efficient approach alongside UCT and random forest model learning (Hester and Stone, 2013). While its goals were similar to ours (real-time planning and learning) TExplore focused on low-level control operations and fully observable domains.

Dimensionality reduction techniques for MDP's such as state aggregation and abstraction (Singh et al., 1995) map elements from a large state set to a set with lower cardinality, but reducing planning time while maintaining performance is challenging. These techniques are generally associated with hierarchical MDP planners, which use domain knowledge to devise and solve intermediate goals contributing to faster convergence (eg. in POMDP's (Vien and Toussaint, 2015)). PolCA and PolCA+ (Pineau et al., 2003) combine state abstraction and hierarchical planning and solve some higher-level tasks in MDP's and POMDP's respectively.

Point-Based Value Iteration is a point-based approximation to simplify belief estimation in POMDP's, which assumes groups of beliefs share the same action choice and therefore, their values too (Pineau et al., 2006). Beliefs may be chosen following policies based on eg. probability or distance. This directly addresses belief-space complexity but using fixed criteria for clusters might have limitations when generalizing across tasks.

POMCP, based on UCT, produced promising results for online planning in complex domains without a complete transition model (Silver and Veness, 2010), but as far as we know it hasn't been applied in higher-level robot or agent planning where external features may affect the real value of states and actions.

MDP planning overlaps with reinforcement learning (RL), a set of techniques to learn an MDP and its optimal policy based solely on perceived numerical rewards. Attempts to speed up convergence in RL include function approximation (cf. (Sutton and Barto, 2012)), using STRIPS plans as domain knowledge (Grzes and Kudenko, 2008) and reward shaping based on a proximity-to-goal heuristic (not always be available or easy to formalize). Potential-based reward shaping however has been shown to preserve policy optimality (Ng et al., 1999). Pure RL approaches are capable of learning complex domains and policies but quickly become intractable and impractical.

Recent results in robot control combined classical planning with assumptions derived from different levels of domain knowledge, managing uncertainty by constructing and solving very small POMDP's (Hanheide et al., 2015). This uncertainty referred to the presence of objects in the environment, so actual problem solving relied on conventional methods.

We can conclude existing approaches in planning and problem solving under uncertainty assume domains are stochastic and partially observable in their entirety and do not take advantage of potential simplifications derived from the actual *problem* represented by the POMDP. As planning domains become more variable and tasks more abstract, the underlying POMDP's will also increase in complexity.

3 MDP'S

A Markov Decision Process is a tuple $\langle S, A, T, \gamma, R \rangle$ where S is a set of states, A is a set of actions, $T : S \times A \times S \to [0,1]$ is a state transition probability function such that $T(s,a,s') = p(s'|s,a)$, $\gamma \in [0,1]$ is a discount factor which determines the horizon, and R is a set of real-valued rewards (or costs) associated with each transition, (s,a,s'). When S, A and R are finite, the MDP is finite. A solution is a policy $\pi : S \to A$ which maximizes the expected sum of rewards. States represent the information available to an agent at a given moment, which may include immediate sensory information as well as that of previous states. If a state carries all relevant information (for action selection), it has the Markov property.

MDP's assume *complete observability*: the agent always knows the true state of the world. In *partially observable* MDP's, an agent has a set Ω of observations and an observation function $O : S \times A \rightarrow \Omega$, where $O(s,a,\omega) = p(\omega|s,a)$ is the probability of observing ω in state s after executing action a. Because one observation could lead to potentially many states and, consequently, poor policies, the agent maintains an internal belief state $b \in B$ and a probability $b(s)$ that s is the current state, where $b_t(s) = Pr(s_t = s \,|\, h_t)$ and $h_t = (a_0, \omega_1, \ldots, a_{t-1}, \omega_t)$ is the *history*, or sequence of actions and observations at time t. A POMDP is therefore a tuple $\langle S, A, T, \gamma, R, \Omega, O \rangle$.

4 TASK PLANNING IN ROBOTS

Planning agents often find constraints and limitations that effectively modify their utility functions, and make optimal behavior impractical. Instead, satisficing behavior involves quickly filtering out action prospects and assessing which elements in the current state might contribute to reaching the goal, an idea inspired by the intuitive notion of *relevance*. When two agents communicate, new information may be considered relevant if some contextual assumption is strengthened by this new information as long as *not much* effort is involved (Sperber and Wilson, 1995), suggesting a notion of context-sensitive utility or cost. A formal approach to relevance should consider a combination of an agent's context (observations, state) and goal, and evaluate its immediate options (actions) with respect to their contribution to solving that particular goal. We can therefore understand relevance as an attentional filter guiding an agent's perception and action selection, implemented through operators or functions, and leading to a series of simplifications. This idea guides our proposed planning methodology.

4.1 Requirements of practical planning

Practical problem solving agents and robots, behave in domains with the following characteristics:

- Multiple sources of uncertainty: non-deterministic actions, inaccurate sensing.
- Potentially large action and state sets.
- Dynamic, changing environments.
- Limited resources (time, information, etc.)

In terms of granularity, we are interested in planning for higher-level tasks that contribute directly to problem solving. This means actions might be somewhat abstract (but eventually grounded) and goals might be loosely defined (i.e. "bring coffee mug" instead of "move to point (x,y)").

Existing approaches address some of these concerns, but aren't yet satisfactory for real-time planning. We argue that an understanding of (PO)MDP solvers should be reached, following "common sense" insight such as avoiding deep probabilistic search for well-known, mundane tasks, and quickly pruning available actions. We propose the following assumptions:

- Practical determinism: sufficiently reliable transitions may be assumed to be deterministic for practical purposes.
- Mixed observability: observations may resemble known states with sufficient confidence (become fully observable).
- Partial solutions: goals may be decomposed into subproblems with known solutions, retrievable from a model. Segments of the MDP may then be solved quickly using a known plan.

These features require mechanisms to quickly estimate state and action values, through a combination of sampling, simulation and knowledge representation. From now on we will refer only to POMDP's given their generality.

4.2 Assumptions and simplifications

We will now develop our planning assumptions. The result is a combination of deterministic and non-deterministic transitions with mixed observability. These two features plus the assumption of a domain model satisfy the aforementioned requirements of practical problem solving. For the following, let S be a finite state set, A be a finite action set, R be a finite set of rewards, B a finite set of beliefs and Ω a finite set of observations.

Determinism. The transition $\tau(s,a)$ is defined in equation 1, which assumes *practical determinism* if the next state transition is known or reliable. Otherwise it follows the usual stochastic transition behavior of regular POMDP's.

$$\tau(s,a) = \begin{cases} s_\tau \in S & \text{iff } \textit{reliable} \text{ or } \textit{known} \\ \omega \in S \text{ with } p(\omega|s,a) & \text{otherwise} \end{cases} \quad (1)$$

Definition 1. *A transition is reliable if* $\exists s' \in S \,.\, p(s'|s,a) \geqslant T_\tau \wedge \nexists s'' \in S \,.\, p(s''|s,a) \geqslant p(s'|s,a)$.

That is, the transition $s' \leftarrow (s,a)$ is assumed deterministic if its probability is at least T_τ, and only one state satisfies this condition.

Definition 2. *A known transition is one retrieved from a model or knowledge representation, with or without estimating its probability.*

Observability. Certain contextual features might affect the current belief distribution enough to directly map it to a state. Such belief states are *well-founded*. The underlying motivation is to circumvent expensive belief computation by using knowledge about the regularity of problems, the domain and their solutions.

Definition 3. *An observation* $\omega \in \Omega$ *is well-identified if it can be mapped to a state* $s \in S$ *by a function* $f : \Omega \rightarrow S$.

We can assume more than one observation may be mapped to some particular state. We may say observations $\omega_1, \ldots, \omega_n \in \Omega$ for $n < |\Omega|$ are associated to state $s \in S \iff f(\omega_1) = \ldots = f(\omega_n) = s$. Finally we define *well-founded* beliefs.

Definition 4. *A belief* $b_t \in B$ *is well-founded if* $b_t\big(f(\omega_t)\big) \geqslant T_\beta$.

which makes the belief of a particular state meet probability threshold T_β, reducing a belief distribution to a state if the current observation is well-identified, i.e. there is reason to assume the current state is, in fact, some $s \in S$. In cases where no such relationship exists and actions to reduce uncertainty must be taken, a conventional solver may be used, albeit in a sufficiently small POMDP.

Hierarchy. Partial solutions are supported by a combination of a model, some form of state aggregation and subtask identification. Explicit domain knowledge may also allow a generative model to make more accurate predictions, thus improving action selection. This idea is similar to hierarchical planning.

These assumptions yield an augmented POMDP $\langle S, A, \tau, \gamma, R, \Omega, O, M \rangle$, where τ is the new transition function and M is a domain model. This POMDP has *only some* non-observable states and *only some* non-deterministic transitions, allowing a robot to consider transitions such as "driving forward" or "going through the door" as *reliable*, beliefs such as "the blob on the door being the handle" as *well-founded* and tasks such as turning the door handle as already solved if a solution exists. We now propose a planning algorithm that makes use of these assumptions.

4.3 Towards practical planning

Our proposed methodology consists of 1) Context-sensitive dimensionality reduction (through abstraction and task hierarchy), 2) Simplified planning and learning (efficient sampling and simulation with careful value backups) and 3) Satisficing solutions. We will now quickly develop these arguments and present our preliminary algorithm for planning under uncertainty.

Relevance functions and operators: A relevance function applies a series of operators to a set (eg. of actions, of observations, of states) to filter out those that don't contribute *sufficiently* to reaching the current goal. Given a set of features K, a set of descriptions or context D and a goal g, relevance function $r : K \rightarrow K_r$ produces a subset K_r such that $|K_r| << |K|$ where $\forall k \in K_r$ k contributes to achieving g. A promising direction is goal-directed simulation with a black-box generator, using an aggregated representation of states or beliefs.

Action selection: Based on the idea of practical determinism for task-planning in POMDP's, the action selection policy may use a generative model to sample likely or known state transitions and find valuable actions. Known action selection policies such as ε-greedy, soft-max or even UCB1 may be followed, using approximate action values on a set containing only relevant entries.

Planning and learning: The overarching algorithm may implement planning alongside a long-term learning rule to incorporate perceived experience, similar to the Dyna family of algorithms (Sutton and Barto, 2012). States or beliefs receiving value updates must be carefully chosen and the number of updates minimized, following relevance-based criteria.

Algorithm 1 formalizes our task planning methodology. Relevance-driven steps are action selection, observation to state mapping, state aggregation and value back ups. A domain model is maintained and updated (when necessary) by the UPDATEM function, and the current best policy is made available to interleave planning and acting. Figure 1 shows the functional cycle, where solid lines represent necessary relationships for execution, dashed lines represent conditional relationships, and dotted lines represent on-demand requests. From the domain model M state and action values may be derived for a generative model and the overall planner. The task planner deliberates over transitions, updates the model and outputs a policy.

This completes our proposal to handle the aforementioned challenges of practical task planning. We address dimensionality and uncertainty through rele-

Algorithm 1 Agent Task Planning

```
Input: initial state $s$, model $M$, goal description $g$
Output: policy $\pi$
function TaskPlanning
    repeat
        $a \leftarrow A(s)$                              ▷ eg. generative model
        $\omega, r \leftarrow (s,a)$
        $s' \leftarrow \begin{cases} (s,a) & \text{iff } \textit{reliable} \text{ or } \textit{known} \\ \omega & \text{iff } \textit{well-identified} \\ \text{PO subproblem} & \text{otherwise} \end{cases}$
        $s' \leftarrow$ aggregated $s'$
        $M \leftarrow$ updateM$(M, s, a, \omega, r, g)$
        Communicate current policy
        $s \leftarrow s'$
    until $s$ is terminal
    return $\pi$
end function

function updateM($M$, $s$, $a$, $\omega$, $r$, $g$)
    Update $M$ with $(s, a, \omega)$
    Update values
    for all $\tau$ in $M$ relevant to $(s, a, g)$ do
        Update associated $s', \omega', r'$
    end for
    return $M$
end function
```

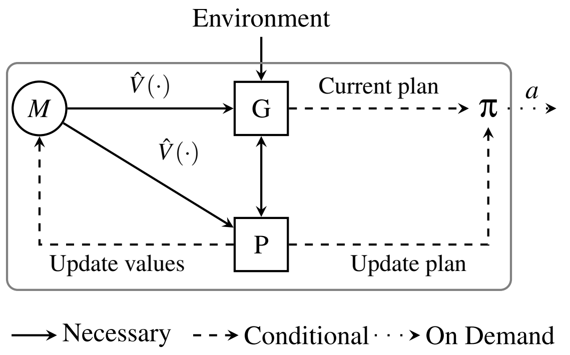

Figure 1: Functional view of the task planner.

vance driven abstraction and contextual assumptions. Transitions with uncertainty that cannot be simplified are solved as regular but small POMDP. Finally, despite not considering external factors explicitly, the agent might still cope with a certain degree of change by interleaving planning and action.

5 DISCUSSION & CONCLUSIONS

We have presented a methodology for planning under uncertainty, with the aim of reducing the complexity of decision processes in robots and other agents. This addresses an open challenge in (PO)MDP planning, in order to control robots capable of real-time problem solving with manipulation. In domains where robots must respond quickly and efficiently, further simplifications are necessary to achieve satisficing behavior. Humans often rely on several cognitive shortcuts such as *insight* and *relevance*, both intuitive and hard to formalize. We have proposed that *relevance* may be understood as context-sensitive, dimensionality reducing operators, in the context of planning and learning.

While domains such as *RockSample* (Smith and Simmons, 2004) serve as benchmarks for POMDP planning algorithms, they do not accurately represent practical problem-solving situations. In a real-world rock sampling problem, the robot may have access to knowledge about the location, shape or appearance of valuable rocks before even sampling them, and other factors such as dangerous terrain might affect the real value of samples. Planning for “simpler” tasks such as finding a cup of coffee may also benefit from practical assumptions. We expect relevance operators will contribute to planning in domains where external factors may introduce biases to quickly achieve satisficing behavior.

Variations of RockSample may also be of interest. For instance, a setting with objects valued according to features (eg. color, size, etc.) correlated with eg. location, and with time and resource limitations (eg. battery life). Observed features would lead to multiple belief distributions, simplified using relevance-based criteria. Information gathering actions are expensive and may sometimes be avoided. When resources or time start running out, the context and utility functions may change: simpler objects might be more valuable than distant, risky ones. A restaurant domain such as that of RACE is also of interest (Hertzberg et al., 2014). Here the robot must keep the patrons’ coffee cups or wine glasses full and like in RockSample, each table yields positive (needs refill) or negative (doesn’t need refill) reward when visited, and checking is costly.

A more complex scenario involves multiple objects and actions, in a puzzle-like setting. Relatively simple goals such as “leave the room” might be tricky if obstacles are present, requiring reasoning over multiple interactions such as pushing or even climbing. State descriptions in this example must be heavily filtered to maintain only relevant features and actions: the location of a small table might be relevant if one can climb on it, or the location of a chair if it can be pushed out of the way. Informative actions might be necessary to assess whether these actions are at all possible.

Evaluation metrics such as cummulative reward, *regret* or computing time may be applied in planning robots but given our interest in abstraction and simplification, analyzing the scalability of these methods in larger domains will also be important. Note that most dimensionality reduction approaches substitute the state set or the evaluation functions in (PO)MDP's and are therefore subject to different convergence and optimality criteria. Our proposal is similar, assuming that complex problems in POMDP form (P) have a simpler, underlying representation (P') from which solutions may be extracted. These solutions should be near-optimal within provable limits for P', so an additional challenge is finding what form of relevance functions and operators preserve these properties when transferring policies back to P. We expect the relevance thresholds previously introduced will allow us to estimate the approximation error.

Since this paper outlines a research project, there is much work yet to be done. The core of this methodology are the context-sensitive relevance functions and operators. A fully-functional system will require state, observation and belief estimation and aggregation. Efficient action selection, through simulation techniques, might be a key step in avoiding irrelevant transitions. Finally, a domain model binds these modules together and supports the practical assumptions. Putting it all together is a challenge in its own right but using context-sensitive criteria is the main innovation of our proposal.

ACKNOWLEDGEMENTS

This work is supported by a DAAD research grant.

REFERENCES

Boutilier, C., Dean, T., and Hanks, S. (1996). Planning under uncertainty: Structural assumptions and computational leverage. In *In Proc. 2nd Eur. Worksh. Planning*, pages 157–171. IOS Press.

Ghallab, M., Nau, D., and Traverso, P. (2016). *Automated Planning and Acting*. Cambridge University Press, San Francisco, CA, USA.

Grzes, M. and Kudenko, D. (2008). Plan-based reward shaping for reinforcement learning. In *Intelligent Systems, 2008. IS '08. 4th Intl. IEEE Conf.*, volume 2, pages 10–22–10–29.

Hanheide, M., Göbelbecker, M., Horn, G. S., Pronobis, A., Sjöö, K., Aydemir, A., Jensfelt, P., Gretton, C., Dearden, R., Janicek, M., Zender, H., Kruijff, G.-J., Hawes, N., and Wyatt, J. L. (2015). Robot task planning and explanation in open and uncertain worlds. *Artificial Intelligence*, pages –.

Hertzberg, J., Zhang, J., Zhang, L., Rockel, S., Neumann, B., Lehmann, J., Dubba, K. S. R., Cohn, A. G., Saffiotti, A., Pecora, F., Mansouri, M., Konečný, Š., Günther, M., Stock, S., Lopes, L. S., Oliveira, M., Lim, G. H., Kasaei, H., Mokhtari, V., Hotz, L., and Bohlken, W. (2014). The RACE project. *KI - Künstliche Intelligenz*, 28(4):297–304.

Hester, T. and Stone, P. (2013). TEXPLORE: Real-time sample-efficient reinforcement learning for robots. *Machine Learning*, 90(3).

Kearns, M., Mansour, Y., and Ng, A. Y. (2002). A Sparse Sampling Algorithm for Near-Optimal Planning in Large Markov Decision Processes. *Mach. Learn.*, 49(2-3):193–208.

Kocsis, L. and Szepesvári, C. (2006). Bandit based Monte-Carlo Planning. In *ECML-06*, pages 282–293. Springer.

Kushmerick, N., Hanks, S., and Weld, D. (1994). An algorithm for probabilistic least-commitment planning. In *AAAI-94*, pages 1073–1078.

Ng, A. Y., Harada, D., and Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. In *In Proc. 16th Intl. Conf. Mach. Learn.*, pages 278–287. Morgan Kaufmann.

Pineau, J., Gordon, G., and Thrun, S. (2003). Policy-contingent abstraction for robust robot control. In *Proc. 19th Conf. on Uncertainty in Artificial Intelligence*, UAI'03, pages 477–484, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Pineau, J., Gordon, G. J., and Thrun, S. (2006). Anytime point-based approximations for large POMDPs. *Journal of Artificial Intelligence Research*, 27:335–380.

Silver, D. and Veness, J. (2010). Monte-Carlo Planning in Large POMDPs. In *In Advances in Neural Information Processing Systems 23*, pages 2164–2172.

Singh, S. P., Jaakkola, T., and Jordan, M. I. (1995). Reinforcement learning with soft state aggregation. In Tesauro, G., Touretzky, D. S., and Leen, T. K., editors, *Advances in Neural Information Processing Systems 7*, pages 361–368. MIT Press.

Smith, T. and Simmons, R. (2004). Heuristic Search Value Iteration for POMDPs. In *Proc. 20th Conf. on Uncertainty in Artificial Intelligence*, UAI '04, pages 520–527, Arlington, Virginia, United States. AUAI Press.

Sperber, D. and Wilson, D. (1995). *Relevance: Communication and Cognition*. Blackwell Publishers, Cambridge, MA, USA, 2nd edition.

Sutton, R. S. and Barto, A. G. (2012). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA, 2nd edition. (to be published).

Thiébaux, S. and Hertzberg, J. (1992). A semi-reactive planner based on a possible models action formalization. In *Artificial Intelligence Planning Systems: Proc. 1st Intl. Conf. (AIPS92)*, pages 228–235. Morgan Kaufmann.

Vien, N. A. and Toussaint, M. (2015). Hierarchical Monte-Carlo Planning. In *AAAI-15*.

von Neumann, J. and Morgenstern, O. (1944). *Theory of Games and Economic Behavior*. Princeton University Press.