Bias in Planning Algorithms

Alison Paredes[1,2] **J. Benton**[1] **Jeremy Frank**[1] **Christian Muise**[2]

[1]Intelligent Systems Division, NASA Ames Research Center
[2]Department of Computer Science, Queens University
Authors listed in arbitrary order.

Abstract

Does bias exist in planning algorithms? If so, how does bias manifest, and how important is this bias? Answering this question requires a formal, mathematical definition of bias. We formally define bias as the distance between the probability distributions of solutions returned by various algorithms, and and the uniform distribution over solutions. We show in this paper that deterministic algorithms are *inherently* biased, as they don't return all solutions, and that this property holds even when algorithms return a *set of plans* instead of just one plan. Exceptions are problem instances or problem classes for which only a *single solution* exists. We then discuss changing the definition of bias to compare the probability distributions of *properties* of sets of plans instead of individual plans. We show the property bias is *smaller* than the bias of actual plans. Finally, we show that *entropy* is a proxy for the more complex and more expensive distance measurement between pairs of probability distributions. We then describe a roadmap for future investigations of bias in planning.

1 Is *Your* Planner Biased?

The problems of bias in AI applications of machine learning has been the subject of considerable debate and recent research (Hort et al. 2023). Many of these biases arise due to biases in the data used as input to these algorithms. Applications like planning may exhibit bias due to unfair rules or optimization criteria, as opposed to due to training data. However, when considering the set of solutions to planning problems, and the possibility that users want to explore multiple solutions to a problem, a different form of bias in *algorithms* becomes a pertinent problem to explore. Recent work (Paredes 2023) has shown that planning algorithms have biases that lead to errors when using these planners to generate sample plans as input to learning systems. To date, however, there has not been a systematic examination of what bias means, and how problematic it is. This paper provides a starting point for discussing bias in planning in a principled way.

We start by defining bias for planning problems requiring one solution. We then move on to define bias for planning problems requiring sets of plans.

Sets of plans are typically required when human users of planning systems have some uncertainty about exactly they want. We will develop most of the foundation with optimal planning problems, because we want to define bias for problems in which a set of plans constitute candidate solutions, and optimality criteria are often used to define sets of plans returned.

2 Planning Problems

We start with defining satisficing planning problems.

Definition 1. *Let* $M= \langle V,O\rangle$ *be a set of variables and planning operators. Let* I *be an initial state consisting of a complete assignment to the variables. Let* G *be a set of goal states implicitly defined by an incomplete assignment to the variables. Let* $\Pi= \langle M, I, G\rangle$ *be a planning problem instance. Let* $S(\Pi)$ *be the space of solutions to* Π *i.e. a (possibly partially) ordered series of operators that achieves the goal from the initial state. Let* $s\in S(\Pi)$ *be a solution.*

Most of our results will be for optimal planning which requires a cost function over individual plans, which is our next definition. We focus on Top-K (Katz et al. 2018) and Top-Quality (Katz, Sohrabi, and Udrea 2020) problems first. We present a brief aside on multi-criteria planning and Pareto problems later.

Definition 2. *Given a planning problem instance* Π. *Let* $F(s){:}S(\Pi)\Rightarrow \mathbb{R}$ *be a cost function. Optimal solutions are the set of plans* $\arg\min_{s\in S(\Pi)} F(s)$. *Let* $\Pi_o= \langle M, I, G, F\rangle$ *be an optimal planning problem instance. Inputs for Top-K planning are the number of plans* k*, and a solution is a set* S *such that* $|S| = k$ *with the property that if* $s \in S$ *then all* r *such that* $F(r) \leq F(s)$ *also in* S *(up to the set size). Inputs for Top-Quality planning are a cost bound* q*, and the solution is the (unique) set* $S = r|F(r) \leq q$.

Diverse-Planning (Srivastava et al. 2007) is the last problem we investigate for bias. Diverse planning requires a distance measure $\delta(r, s)$ measuring the 'distance' between two plans. Plan diversity does not strictly require an optimization criteria, but diversity could be measured over plan cost (or costs, if the problem is a multi-criteria optimization problem), and bounds on plan cost can be used to limit those plans that otherwise must be diverse. While there are many possible variants of diverse planning, we start with a simple one to illustrate the issues of bias in a similar way to those we have discussed above.

Definition 3. *Let Π_o be an optimal planning problem instance. Let $\delta(r,s)$ be a measure of the distance between two feasible plans, i.e. $\delta(r,s)$: $S(\Pi)\times S(\Pi) \Rightarrow \mathbb{R}$. Then a Diverse-Planning problem instance is a tuple $\langle M,I,G,F,\delta\rangle$. Inputs are set size k and pairwise distance bound b, i.e. $|S|=k$ and all pairs of plans (r,s) in the solution S must have the property $\delta(r,s)\geq b$.*

3 A Basis for Bias

We start by defining bias for planning problems requiring one solution. We use the term distance measure over probability distributions below. Some such measures are not metrics because they don't obey the Triangle inequality (e.g. Bhattacharya distance). The exact measure or metric used is unimportant at this time, but for later results, we will use a specific distance measurement.

3.1 Plan Bias

Definition 4 (Plan Bias)**.** *Let $\Pi_o=\langle M,I,G,F\rangle$ be an optimal planning problem instance. Let A_1 be an algorithm returning a single solution. Let $P_{i,\Pi_o}(s)$ be the probability that Algorithm A_i generates solution s to Π_o. Define probability distribution $P_{0,\Pi_o}(s)=\frac{1}{|S(\Pi_o)|}$, that is, each solution is returned assuming plans in $S(\Pi_o)$ are sampled uniformly at random. Let $\Delta(P_{1,\Pi_o},P_{2,\Pi_o})$ be a distance measure over probability distributions. Then A_i is* biased *if $\Delta(P_{0,\Pi_o},P_{i,\Pi_o})>0$.*

Lemma 1. *Let A_i be a deterministic planning algorithm. Then A_i is biased if and only if $|S(\Pi_o)|\geq 1$.*

Proof. A deterministic algorithm A_i returns the same solution s given the same inputs; equivalently, $P_{i,\Pi_o}(s)=1$ for some $s\in S(\Pi_o)$ and $P_{i,\Pi_o}(r)=0$ for all other solutions. Therefore $|S(\Pi_o)|>1 \Rightarrow \Delta(P_{0,\Pi_o},P_{i,\Pi_o})>0$. Otherwise, $\Delta(P_{0,\Pi_o},P_{i,\Pi_o})=0$. □

Lemma 2. *Let A_e be the (EXPTIME) algorithm that enumerates all optimal solutions in $S(\Pi_o)$ and samples from among them u.a.r. Then A_e is unbiased.*

Proof. Trivial. (The algorithm is EXPTIME and not EXPSPACE because we can count the number of solutions without saving them, decide which should be returned, then generate them again and count solutions, returning the proper one when it is generated.) □

Lemma 3. *Let A_i be a deterministic algorithm and A_j be a non-deterministic algorithm. Then $\Delta(P_{0,\Pi_o},P_{i,\Pi_o})\geq \Delta(P_{0,\Pi_o},P_{j,\Pi_o})$.*

Proof. Trivial because any non-deterministic algorithm must return at least one plan. □

Lemmas 1,2 and 3 indicate that bias, as we have defined it, does not seem to be an interesting question to address for algorithms that return a single plan, because *all* deterministic algorithms exhibit bias, and unbiased algorithms require enumerating all plans, and thus exponential time. Even were we to extend the analysis above to multiple planning problem instances, where deterministic bias would now identify different individual plans solving different instances, it's not clear what is gleaned from the observation of bias. Bias is of interest for non-deterministic algorithms, especially in relation to each other, and will be discussed later.

3.2 Plan Set Bias

We now turn our attention to algorithms that return multiple plans. Some of these set-planning problems are posed in such a way as to ensure a 'fair sample' of the solutions are provided. If such algorithms are biased, this is a problem to be solved. As we will show, bias does indeed exist in algorithms to solve such problems.

Three common optimal planning problems that require sets of solutions are Top-K, Top-Quality, and Plan-Diversity. Top-K and Plan Diversity both ask for sets of k plans, while Top-Quality asks for all plans whose quality exceeds a bound; in our cost minimization setting, Top-Quality asks for all plans whose cost is lower than q. We have crafted our definitions so that our algorithms can return sets of plans, and we can measure bias of these sets of plans compares to all possible sets of plans satisfying the problem description.

Definition 5 (Plan Set Bias)**.** *Let Π_o be a set-planning problem instance, consisting of $\langle M,I,G,F\rangle$ combined with additional inputs Σ implicitly defining* sets of *solutions as outputs. Specifically, $\Sigma\in\{k,q,(\delta,b,k)\}$, defining either a Top-K, Top-Quality or Plan Diversity problem. Let $S(\Pi_o,\Sigma)\subset 2^{S(\Pi)}$ be the* set of solution sets *to a set-problem instance. Let A_i be an algorithm returning a solution set. Let $P_{i,\Pi_o}(S)$ be the probability that Algorithm A_i generates solution set $S\in S(\Pi_o,\Sigma)$. Define probability distribution $P_{0,\Pi_o}(S)=\frac{1}{|S(\Pi_o,\Sigma)|}$, that is, each solution set is returned assuming $S\subset S(\Pi_o,\Sigma)$, are sampled uniformly at random. Let $\Delta(P_{1,\Pi_o},P_{2,\Pi_o})$ be a metric over probability distributions. Then A_i is* biased *if $\Delta(P_{0,\Pi_o},P_{i,\Pi_o})>0$.*

Lemmas 1 - 3 still apply, i.e. if a set-problem has exactly one solution plan set, deterministic algorithms aren't biased, but if a set-problem admits multiple sets as answers, then a deterministic algorithm is biased, and non-deterministic algorithms are less biased than deterministic algorithms.

Once we investigate the details of different set-problems, we find we can make some more precise statements about the existence of bias, as we will see in the results below:

Theorem 1. *Let Π_o be an optimal planning problem instance. Let A_i be a sound and complete Top-Quality algorithm with a cost bound q as input. Then A_i is unbiased.*

Proof. By definition, there is only one set of plans whose costs are all $\leq q$, so any sound and complete Top-Quality must return this set. □

Theorem 2. *Let Π_o be an optimal planning problem instance. Let A_i be a deterministic Top-K algorithm with a set size k as input. Let S' be a solution set to the Top-K problem defined by Π_o and k. Let $f=\max_{s\in S'}F(s)$, that is, f is the maximum cost of any plan in any solution to the Top-K problem instance. Let $s_f=|s\in S' s.t. F(s)=f|$, that is, the number of maximum cost plans in any solution to the Top-K instance. Let $n_f=|s\in S(\Pi) s.t. F(s)=f|$, that* *is, the number of feasible solutions to the optimal planning problem instance derived from Π_o whose cost is f. Then A_i is biased if and only if $n_f > s_f$.*

Proof. The worst-case cost f of a solution to a Top-K problem is a function of k the size of the set requested, the feasible plans in the domain, and the cost function. The number of plans of this worst-case cost in a solution is also a function of k and the feasible plans in the domain. The number of plans in Π_o, however, is not dependent on the size of the set k, but is rather a function of the instance itself. If there are enough worst-cost plans that, for a given k, the Top-K algorithm can choose some of those worst-case cost plans to fill out a solution set S', then any deterministic algorithm will return only one of the many sets that could be constructed. The remainder of the proof follows similar lines to the other deterministic bias results. □

Deterministic bias of Top-K algorithms appears to be a less-problematic form of bias, because the bias exists only in the choice of the 'worst' plans in the set. Bias is still potentially a problem if the difference in quality between the optimal plan and the worst plan in the set is small. However, the problem of bias can also be essentially eliminated if we ask for a Top-Quality solution as opposed to a Top-K solution.

Theorem 2 shows that Top-K problems have an unintuitive property. We don't necessarily know the distribution of $F(s)$, and this n_f, up-front. While it seems reasonable that there are increasing numbers of plans with larger $F(s)$, it's not a guarantee, and so the sizes of the sets correspoding to solutions of Top-K problems aren't known either. Put another way, the set size requested isn't a 'constraint' in the usual way one would expect it to be.

As an aside, the definitions of bias can be extended to multi-criteria problems and algorithms that find the Pareto Frontier. The Pareto Frontier is unique, so sound and complete algorithms will be unbiased. Approximations of the Pareto Frontier may be of interest but the definitional machinery needed is probably not worth investing in at this time. Analogs of 'Top-K' could also be defined, i.e. take the union of Top-K for each cost function separately; now we have biased deterministic algorithms.

In general, deterministic algorithms for Diverse-Planning will exhibit bias when there are multiple sets of k plans such that all pairs of plans in the solution S must have the property $\delta(r, s) \geq b$, leading to the following result:

Theorem 3. *Let $\langle M, I, G, F, \delta \rangle$ be a Diverse-Planning problem instance. Let A_i be a deterministic Diverse-Planning problem with input Σ consisting of set size k and pairwise distance bound b. Let Π_o be a solveable diverse planning problem instance for set size k and pairwise distance bound b. Then there exists $k' \leq k$ and $b' \leq b$ such that A_i is biased.*

Proof. Let $S, \in S(\Pi_o, \Sigma)$. Let $m \leq n$, and let $S' \subset S$ be formed by arbitrarily removing m plans from S. Let $b' = \min_S \min_{r,s \in S} \delta(r, s)$. Trivially $b' \geq b$ since S solves the diverse planning problem instance Π_o with inputs b, k; removing one or more plans from S could increase the largest distance in this solution, but when we remove m plans from the set we trivially obtain $\binom{k}{m}$ solutions to a problem with set inputs $k' = k - m, b'$. □

Diverse Planning seems like a good target for investigating bias because, as defined, problem instances can have many solutions, and any algorithm will need to choose (arbitrarily) one of the many solutions. Unlike single plan problems, the set will consist of many plans; unlike Top-K, the free choice in the set of plans is not necessarily limited to the poorest quality plans.

This manifestation of bias in Diverse-Planning algorithms can be slightly mitigated by finding the largest k or the largest b for which a solution exists. However, there may still be multiple solutions to the problem defined by the largest set, so deterministic algorithm bias may still exist.

4 Changing the Bias

The definition of bias we have used so far is very strict. Bias exists if algorithms do not return plans in the space of solutions to the satisficing planning problem instance. We propose to focus on *plan property bias* instead of individual plan bias. The idea is that instead of measuring bias in the sets of plans that are produced compared to the sets of feasible plans, we instead focus on the properties of sets of plans. If many plans, or plan sets, are similar because they have similar properties, then bias in plan sets may not translate to bias in the properties of interest. Instead, it may make more sense to *summarize* or *abstract* a set of plans by the properties exhibited by plans in the set. As we will see in later sections, there is also justification in using properties to define the cost functions and distance metrics used to define our set-planning problems, and asking if there is bias in these properties, rather than the plans themselves.

What properties might we want to use? The goal propositions in the satisficing planning problem aren't suitable, but if goals have value, then achieved goal propositions would be useful properties. We could use a set of propositions not in the goals that are considered 'interesting', i.e. side effects like resource consumption. We could use the cost function, or for Pareto problems, the cost functions. We could use the plan diversity distances. We could use propositions as 'trajectories', i.e. states experienced during the plan. We also hope to reduce the bias we see in deterministic algorithms by focusing on aspects of plans that are important rather than the actual plans. If a large set of plans is summarized by a small set of properties, then perhaps opportunities for bias are minimized.

4.1 Property Bias

To define property bias, we first need a mapping of feasible plans to properties, and a mapping from plan sets to property sets. We then need to translate plan set probability distributions into property set distributions. Then our set-planning problems produce sets of plans that also map to property sets, and we can proceed as we did before, and measure the distance between the probability distributions when plan sets are returned u.a.r. versus the algorithm probabilities.

Definition 6 (Property)**.** *Let Π_o be a planning problem instance, consisting of $\langle M, I, G, F\rangle$ Let ϕ_i be a property of a solution $s\in S(\Pi_o)$. We will abuse notation and treat ϕ_i as a Boolean function mapping plans to true or false, $\phi_i(s) = \top$ if s has property ϕ_i and $\phi_i(s) = \bot$ otherwise, thus $\phi_i(s)$: $S(\Pi)\Rightarrow \mathbb{B}$.*

A set of solutions may have many plans with a specific property. One option is to ask if two sets of solutions returned by an algorithm have the same properties, or if we want those sets of plans to have not only the same properties, but the same numbers of plans with those properties. To ease the bias burden, we will start with the idea of the set of properties exhibited by *any* plan in the set, and discuss alternatives in later sections.

Definition 7 (Property Set)**.** *Let Π_o be a planning problem instance, consisting of $\langle M, I, G, F\rangle$ Let $\phi_1 \ldots \phi_j$ be properties of a solution $s\in S(\Pi_o)$. Let $S\subset S(\Pi_o)$. Denote the* property set *of S by $\Phi_S= \cup_{i,s\in S}\phi_i(s)$, that is, all properties ϕ_i for which there exists $s\in S$ such that $\phi_i(s) = \top$.*

We will denote the function $\Phi(S)$ that computes Φ_S. If Φ_* is the set of all properties, 2^{Φ_*} is the power set of all property sets that could be exhibited by a plan set (including the empty set).

Definition 8 (Property Set Bias)**.** *Let Π_o be a set-planning problem instance, consisting of $\langle M, I, G, F\rangle$ combined with additional inputs Σ implicitly defining* sets of *solutions as outputs. Let $S(\Pi_o, \Sigma)\subset 2^{S(\Pi)}$ be the* set of solution sets *to set-problem instances.*

Let A_i be an algorithm returning sets of plans as solutions to the set-planning instance. Denote the probability that property set Φ_S is returned by the algorithm by $P_{i,\Pi_o}(\Phi_S)$.

Denote by $S(\Pi, \Sigma, \Phi_S)$ those plan sets in $S(\Pi_o, \Sigma)$ exhibiting property set Φ_S. Define probability distribution $P_{0,\Pi_o}(\Phi_S)= \frac{|S(\Pi,\Sigma,\Phi)|}{|S(\Pi_o,\Sigma)|}$, that is, each property set is returned assuming $S\subset S(\Pi_o, \Sigma)$, are sampled uniformly at random. Let $\Delta(P_{1,\Pi_o}(\Phi(S)), P_{2,\Pi_o}(\Phi(S)))$ be a distance measure over probability distributions. Then A_i is biased *if $\Delta(P_{0,\Pi_o}(\Phi(S)), P_{i,\Pi_o}(\Phi(S))) > 0$.*

There is an analog of Theorem 1 for Top-Quality; since there is still only one Top-Quality plan set, it has a unique property set, so all algorithms exhibit no Property Set Bias. The analogous theorem of Theorem 2 needs to be re-cast in terms of the number of distinct property sets corresponding to Top-K solution sets, but also otherwise holds, as does Theorem 3. So far, it seems that structurally, at least, the use of property sets does not strongly influence the existence of bias.

We suggested that by mapping sets of plans to sets of properties of sets of plans, deterministic algorithms will be penalized less than they would be over sets of actual plans, because the number of distinct property sets would be smaller than the number of distinct plan sets [1]. The intuition is that the 'coarser' probability distributions resulting from mapping many plan sets to the same property set will reduce the distance between the resulting probability distributions. We would also expect to see less bias for property sets with fewer properties, for similar reasons. We prove this result (for one distance measure) below:

Theorem 4. *Let $\langle M, I, G, F\rangle$ be a set-planning problem instance with additional inputs Σ implicitly defining sets of solutions. Let A_i solve this problem. Let Φ_S be a property set over solutions. Then $\Delta(P_{0,\Pi_o}, P_{i,\Pi_o}) \geq \Delta(P_{0,\Pi_o}(\Phi(S)), P_{i,\Pi_o}(\Phi(S)))$.*

Proof. Consider the (discrete) Bhattacharya distance Δ_B, a common measure of 'distance' between two probability distributions. $\Delta_B(P_1(X), P_2(X)) = -\log(\sum_{x\in X} \sqrt{P_1(x)P_2(x)})$. We can investigate the case where, initially, $X = S(\Pi_o, \Sigma)$, the set of plan sets, and after consolidation, they lead to plan property sets $y \in Y$. For simplicity below we will denote $P_{0,\Pi_o}(X)$ by $P(X)$ and $P_{i,\Pi_o}(X)$ by $Q(X)$ and $P_{0,\Pi_o}(\Phi_X)$ by $P(Y)$ and $P_{i,\Pi_o}(\Phi_X)$ by $Q(Y)$.

We want to show that the plan property bias is smaller than the plan set bias, thus we want to show

$$-\log\left(\sum_{y\in Y}\sqrt{P(y)Q(y)}\right) \leq -\log\left(\sum_{x\in X}\sqrt{P(x)Q(x)}\right)$$

Multiplying both sides by -1 yields:

$$\log\left(\sum_{y\in Y}\sqrt{P(y)Q(y)}\right) \geq \log\left(\sum_{x\in X}\sqrt{P(x)Q(x)}\right)$$

Exponentiating both sides yields:

$$\left(\sum_{y\in Y}\sqrt{P(y)Q(y)}\right) \geq \left(\sum_{x\in X}\sqrt{P(x)Q(x)}\right)$$

Let $X_y \subset X$ be those plan sets with the same plan set property corresponding to $y \in Y$. Specifically, the plan property set partitions the plan sets. We now show that for each y

$$\sqrt{P(y)Q(y)} \geq \left(\sum_{x\in X_y}\sqrt{P(x)Q(x)}\right)$$

Squaring both sides yields:

$$P(y)Q(y) \geq \left(\sum_{x_i,x_j\in X_y}\sqrt{P(x_i)Q(x_i)P(x_j)Q(x_j)}\right)$$

Recall $P(x_i)$ is some constant derived from the uniform distribution of plans in $S(\Pi_o)$; denote this by $\frac{1}{\alpha}$. Similarly, $P(y)$ is also derived from the uniform distribution of plans in $S(\Pi_o)$ and the number of plan property sets; denote this

[1] One could hypothetically construct property sets larger than the number of plan sets, but it seems unlikely to be worthwhile to do so in practice.

by $\frac{\beta}{\alpha}$. Specifically, β is the number of plan sets with a given plan property set. After substitution:

$$\frac{\beta}{\alpha}Q(y) \geq \left(\sum_{x_i, x_j \in X_y} \sqrt{\frac{Q(x_i)Q(x_j)}{\alpha^2}} \right)$$

Simplifying:

$$\frac{\beta}{\alpha}Q(y) \geq \left(\frac{1}{\alpha} \sum_{x_i, x_j \in X_y} \sqrt{Q(x_i)Q(x_j)} \right)$$

Recall $Q(y) = \sum_{x_i, x_j \in X_y} Q(x_i)$; making this substitution on the left side yields

$$\frac{\beta}{\alpha}\left(\sum_{x_i \in X_y} Q(x_i) \right) \geq \left(\frac{1}{\alpha} \sum_{x_i, x_j \in X_y} \sqrt{Q(x_i)Q(x_j)} \right)$$

Let $x_m = \arg\max_i Q(x_i)$. Because $Q(x_j) \leq Q(x_m)$ we see that

$$Q(x_m) \geq \sqrt{Q(x_m)Q(x_j)}$$

We can group all terms on the right side with x_m; there are β such terms, and thus

$$\frac{\beta}{\alpha}Q(x_m) \geq \left(\frac{1}{\alpha} \sum_{x_j \in X_y} \sqrt{Q(x_m)Q(x_j)} \right)$$

If we now proceed down the list of x_j to the next largest, we see the same inequality holds (there are, in fact, fewer and fewer mixed terms remaining, so the second largest plan property probability only has $\beta - 1$ mixed terms, the third highest has $\beta - 2$, etc.

Recapping: We have now shown that for each y

$$\sqrt{P(y)Q(y)} \geq \left(\sum_{x \in X_y} \sqrt{P(x)Q(x)} \right)$$

Recalling the inequality

$$\left(\sum_{y \in Y} \sqrt{P(y)Q(y)} \right) \geq \left(\sum_{x \in X} \sqrt{P(x)Q(x)} \right)$$

each term in the left sum for a plan property set y has a corresponding set of terms from the set X_y in the right sum corresponding to for which the inequality holds, thus proving the result. □

We return to our original motivation for focusing on plan set bias. Knowing now that an algorithm's plan bias is larger than or equal to its plan set bias suggests that the plan bias is too 'pessimistic' a measure of the bias. Focusing instead on the properties of sets of plans will provide a more 'accurate' assessment of the bias.

4.2 Property Hierarchies Reduce Bias

Do plan property sets with *fewer* properties lead to smaller bias than property sets with more properties? Consider two *arbitrary* plan property sets. The probabilities over plan property sets induced by arbitrary property sets are also arbitrary. We might select a large set of properties that happen to be disfavored by planners, or a small set of properties that planners fairly sample. However, if we take the *union* of two or more properties to create a new property set, we would expect bias to go down, because the new property set combines the plans from the more fine-grained properties. Let $\Phi_X, \Phi_Y \subset \Phi$ be sets of properties. Basically, each property set partitions the set of plan sets. Suppose Φ_Y has fewer 'buckets' than Φ_X. Suppose further that we guaranteed that some of the 'buckets' in Φ_X are re-allocated to other 'buckets' in Φ_Y to achieve the reduction in granularity, in exactly the same manner that plan sets are 'bucketed' by property sets in the first place. Then we would expect $\Delta(P_{0,\Pi_o}(\Phi_Y(S)), P_{i,\Pi_o}(\Phi_Y(S))) \leq \Delta(P_{0,\Pi_o}(\Phi_X(S)), P_{i,\Pi_o}(\Phi_X(S)))$, for similar reasons to our proof above that plan set bias is always larger than plan property set bias.

Corollary 1. *Let $\langle M, I, G, F \rangle$ be a set-planning problem instance with additional inputs Σ implicitly defining sets of solutions. Let A_i solve this problem. Let $S(\Phi_X)$ be those plan sets in $S(\Pi_o, \Sigma)$ exhibiting property set Φ_X. Suppose we have two property sets Φ_X, Φ_Y such that $|\Phi_X| > |\Phi_Y|$, and for every property set $\Phi_y \subset 2^{\Phi_Y}$ there is a property set $\Phi_x \subset 2^{\Phi_X}$ such that $S(\Phi_x) \subset S(\Phi_y)$. Then $\Delta(P_{0,\Pi_o}(\Phi_Y(S)), P_{i,\Pi_o}(\Phi_Y(S))) \leq \Delta(P_{0,\Pi_o}(\Phi_X(S)), P_{i,\Pi_o}(\Phi_X(S)))$*

Using this observation, we can construct the partially ordered lattice of bias for a given 'basis' of property sets.

4.3 Generalizing the Property Set Functions

We have assumed that the property sets of plan sets are functions of properties of single plans. We also assumed in Definition 7 the property set of a set of plans is the *union* of the properties of the plans. There's no reason to restrict ourselves to unions of the properties of single plans; we can say the property set of a set of plans is the *intersection* of the properties of the plans in a set, or any other set function. None of the results above change; the property sets still partition the plans, and thus the plan sets, meaning Theorem 4 and Corollary 1 still apply.

Furthermore, there is also no reason to restrict the property set functions to be functions over single plans; for plan diversity, as we see in the notes below, we can benefit from generalizing the definition of property functions to functions of pairs of plans. In general, the property set is already defined as a function of the whole set of plans anyway. Again, none of the results above change; now, property sets are defined over sets of plans, but the plan sets are still suitably partitioned. All that changes is the probability calculations.

5 The Cost of Assessing Bias

In the previous development of bias, we used distance measures between the probability of returning a plan or plan set, and the uniform distribution over those plans or plan sets. While theoretically useful in characterizing bias, there are practical problems in using such approaches; we would need to enumerate all solutions to a single problem instance, then perform random sampling on top of this to characterize the probability some algorithm returns each plan. Is there a less expensive way to characterize bias?

Suppose we sampled some number of solutions to estimate the probability distribution P_{i,Π_o} of plans returned by some algorithm A_i. We denote this estimated distribution by $\hat{P}_{i,\Pi_o}$. We could measure the bias of this distribution relative to the uniform distribution over those plans found. We note that in (Paredes 2023), the uniform distribution over plans found is used to create less biased samples of plans, but it is not used to compute bias. This approach is also is subject to the problem that not all plans in $S(\Pi_o, \Sigma)$ are found, and thus some plans will be missing entirely from any calculation of bias performed this way.

An alternative is to compute the *entropy* of this distribution, $H(\hat{P}_{i,\Pi_o}) = -\sum_s \hat{p}_i(s) \log(\hat{p}_i(s))$. We know bias is maximized when algorithms return solutions according to the uniform distribution. We also know entropy is maximized when the distribution is uniform. So while the functional forms might be different (depending on the particular distance measure that is used), as the probabilities change, entropy and bias behave in a similar way. Ideally we would like to prove that if A_i has higher entropy than A_j, we can also conclude that A_i is less biased than A_j. Such a proof depends on the distance metric used for computing bias. While it seems difficult to extract the entropy measure directly from the Bhattacharya distance metric easily, the (asymmetric) Kullbach Liebler Divergence offers a more promising approach to this idea, as the next result shows:

Theorem 5. *Let $\langle M, I, G, F\rangle$ be a set-planning problem instance with additional inputs Σ implicitly defining sets of solutions. Let A_i solve this problem. Let $\Delta_{KL}(P_{i,\Pi_o}||P_{0,\Pi_o}) = \sum_s p_i(s) \log(\frac{p_i(s)}{(\frac{1}{|S(\Pi_o)|})})$ be the KL Divergence of P_{i,Π_o} relative to P_{0,Π_o}. Then $\Delta_{KL}(P_{i,\Pi_o}||P_{0,\Pi_o}) = -H(P_{i,\Pi_o}) - \log(\frac{1}{|S(\Pi_o)|})$.*

Proof. Recall $P_{0,\Pi_o}(S)= \frac{1}{|S(\Pi_o,\Sigma)|}$. We then have:

$$\begin{aligned}
\Delta_{KL}(P_{i,\Pi_o}||P_{0,\Pi_o}) &= \sum_s p_i(s) \log(\frac{p_i(s)}{(\frac{1}{|S(\Pi_o)|})}) \\
&= \sum_s p_i(s) \left(\log(p_i(s) - \log(\frac{1}{|S(\Pi_o)|}) \right) \\
&= \sum_s p_i(s) \log(p_i(s) - \sum_s p_i(s) \log(\frac{1}{|S(\Pi_o)|}) \\
&= -H(P_{i,\Pi_o}) - \log(\frac{1}{|S(\Pi_o)|}) \sum_s p_i(s) \\
&= -H(P_{i,\Pi_o}) - \log(\frac{1}{|S(\Pi_o)|})
\end{aligned}$$

The term $-\log(\frac{1}{|S(\Pi_o)|})$ is positive since log of quantities less than one are negative. The first term is the negative of the entropy of the probability of returning a plan. When that probability is uniform, $\Delta_{KL}(P_{i,\Pi_o}||P_{0,\Pi_o}) = 0$; when it is anything other than uniform, the entropy decreases, the negative of the entropy increases, so $\Delta_{KL}(P_{i,\Pi_o}||P_{0,\Pi_o}) > 0$. □

Notice that we don't need to enumerate all solutions, or even estimate the number of solutions, to compute the entropy. We merely need good estimates of $\hat{P}_{i,\Pi_o}$ given those solutions returned by the non-deterministic algorithm. We also don't need to use the somewhat complex distance measures over probability distributions, and decide which of the options to pick. While solutions will still be missing if we don't guarantee all of them are found, and thus the entropy $H(\hat{P}_{i,\Pi_o}) = -\sum_s \hat{p}_i(s) \log(\hat{p}_i(s))$ will not capture everything, this observation offers a reduced cost method of computing a proxy for bias that is useful, as we discuss further in later sections. Finally, we can extend this calculation to plans or plan sets returned by any number of algorithms. Notice the entropy for algorithm A_i can still be computed if A_j produces a solution that A_i did not produce; the entropy contribution for A_i is zero.

Having established this, we can prove a simplified version of Theorem 4, replacing the more complex metric between probabilities with entropy:

Theorem 6. *Let $\langle M, I, G, F\rangle$ be a set-planning problem instance with additional inputs Σ implicitly defining sets of solutions. Let A_i solve this problem. Let Φ_S be a property set over solutions. Then $H(P_{i,\Pi_o}) \geq H(P_{i,\Pi_o}(\Phi(S))$.*

Proof. As with Theorem 4, to simplify the notation, let X be the set of plan sets, and Y be the set of property sets, and let Y_X denote those plan sets with property set Y, and recall that Y partitions X (every plan set in X is in one partition element y). So we want to prove

$$H(P(X)) \geq H(P(Y))$$

Expanding the entropy formulas:

$$\begin{aligned}
-\sum_{x \in X} p(x) \log(p(x)) &\geq -\sum_{y \in Y} p(y) \log(p(y)) \\
&= -\sum_{y \in Y} \sum_{x \in X_Y} p(x) \log(\sum_{x \in X_Y} p(x)) \\
&= -\sum_{x \in X_Y} p(x) \log(\sum_{z | z, x \in X_Y} p(z))
\end{aligned}$$

The somewhat inelegant condition above captures the fact that, for the entropy of the plan property set distribution, we sum all those plan sets with identical property sets inside the log. We know $p(x) < 1$ and $\sum_{z|z,x \in X_Y} p(z) < 1$, so $\log(p(x)) < 0$ and $\log(\sum_{z|z,x \in X_Y} p(z)) < 0$. We also see that, because $\log()$ is increasing,

$$\log(p(x)) < \log(\sum_{z|z,x \in X_Y} p(z))$$

Therefore, for each x,

$$-p(x)\log(p(x)) > -p(x)\log(p(\sum_{z|z,x\in X_Y} p(z))$$

This means

$$-\sum_{x\in X} p(x)\log(p(x)) \geq -\sum_{x\in X_Y} p(x)\log(\sum_{z|z,x\in X_Y} p(z))$$

completing the proof. □

6 Results: Your Planner is Biased

A summary of results:

- Bias is measured using probability of plan sets or plan property sets.
- Deterministic algorithms producing a single solution to most problems, even those returning sets of plans, are inherently biased, regardless of whether plan sets or plan property sets are used to measure bias. The root cause of this is the number of solutions to problem instances. In limited cases, the definition of the problem itself ensures there is a unique solution, thus there is no possibility of bias.
- Of the set-planning problems we have described, Diverse-Planning is the most interesting in terms of the existence of bias. Top-K planning problems have some possibility of bias, but it is of limited interest (the only arbitrary plans are the poorest quality plans) and the related Top-Quality problem has a unique set solution. Pareto problems also have no bias.
- Property set bias is smaller than plan set bias over the same set of plan sets. Combining a 'ground' set of plan properties to create new properties reduces bias.
- Exponential time enumeration and sampling may be needed to create truly unbiased algorithms.
- Entropy and bias are closely related (especially via distance measures like Δ_{KL}) and offer opportunities to assess bias without the need to enumerate all solutions to problem instances, but evaluating bias likely still takes exponential time.

Our results so far are summarized in the table below.

Problem	Alg	Bias	Notes
Optimal	Det.	Biased	1 set ⇒ Unbiased
Optimal	Non-Det.	TBD	Algorithm dependent
Top-K	Det.	Biased	1 set ⇒ Unbiased
Top-K	Non-Det.	TBD	Algorithm dependent
Top-Q	Det.	Unbiased	1 set by definition
Top-Q	Non-Det.	Unbiased	1 set by definition
Diverse	Det.	Biased	1 set ⇒ Unbiased
Diverse	Non-Det.	TBD	Algorithm dependent

7 Investigating Bias

What do the results above say about further investigations of bias? In this section, we describe a roadmap for future research on this topic.

7.1 Investigating Bias for Deterministic Algorithms

Single-Plan Algorithms: Investigating bias for single-plan algorithms (plan existence) don't appear interesting, because every such algorithm is biased. However, there is still value in comparing different deterministic algorithms to each other, in order to ask how the plans are produced by each algorithm might be biased. This line of investigation doesn't require the heavyweight machinery of probability distance metrics; the probabilities of sets of properties produced by these algorithms can be compared directly. By way of example: suppose one gave 10 planners the same problem instance. Are the same plans produced, or not? Why or why not? Do algorithms with common infrastructure (say, FF heuristics) produce the same plans, or not?

Plan-Set Algorithms: As we discussed, bias in Top-K planning is potentially uninteresting, because the free choice is limited to the worst plans; however, there may still be some interesting investigations when the span of quality is small. A comparison of deterministic Top-K and Top-Quality is also of interest, since Top-Quality is guaranteed to be unbiased. We suggest that, of the various plan-set problems we have discussed, that investigations of bias in deterministic algorithms appear to be of most interest for Diverse-Planning. In this setting, if some sets of plans are systematically ignored in favor of others, then the quest for 'diversity' seems to be in jeopardy. For instance, are pairs plans whose distance is close to b ignored in favor of plans of maximum distance, especially when k is 'small' and many solutions exist?

7.2 Investigating Bias for Nondeterministic Algorithms

Bias in Approximation Algorithms: Investigations of bias in non-deterministic algorithms are valuable for both single-plan and plan-set algorithms. There are two settings in which we can investigate non-deterministic algorithm bias. The first is sound algorithms, i.e those ensuring a solution is found, but for which the solution can vary. Are non-deterministic algorithms still biased? If so, how? And given that the best unbiased algorithm we can imagine takes EXPTIME, can we generate unbiased algorithms with lower expense?

Approximation algorithms for very large or very hard problems pose a different challenge for the investigation of bias. Consider, for instance, a Top-K or Top-Quality problem, in which the optimization problem can't be solved. These algorithms are likely biased, because optimal plans are hard or impossible to find. Assessing the true bias for such algorithms, unfortunately, requires finding those solutions; if this is simply impossible, we may be reduced to comparing the bias using entropy.

Entropy and Bias for Non-deterministic Algorithms (Chan et al. 2014),(Diakonikolas and Kane 2016) show that a number of samples proportional to $\sqrt{|S(\Pi_o)|}$ (and the desired error) are needed to characterize whether an unknown distribution (say, the probability of solutions from a randomized algorithm) is 'close to' a known distribution (say, the uniform distribution). The results use the l_1 norm, but are illustrative; in general, we don't need to enumerate all solutions to our planning problems. However, for a given problem instance, we don't know $|S(\Pi_o)|$, and even if we did, the number of samples needed may still large and costly to obtain. We would like to say: if $H(\hat{P}_{i,\Pi_o}) \leq H(\hat{P}_{j,\Pi_o})$ then $H(P_{i,\Pi_o}) \leq H(P_{j,\Pi_o})$ i.e. the relative order of bias computed using true entropy is the same as the order resulting from the sample entropy. Empirical questions we could investigate include: 1) if we gave some number of randomized algorithms increasing amounts of time, how would their entropies change? 2) How does the entropy estimate compare to the bias estimate if we took the time to actually compute it, assuming that is possible?

7.3 Plan Set vs Property Set Bias

Single-Plan Algorithms: Just as it is sensible to ask whether deterministic algorithms return the same or different plans for the same planning problem instance, we can ask if algorithms return plans with similar or different properties. Since many plans share the same properties, changing the focus to those properties of interest may be more fruitful in the single plan algorithm case, regardless of whether the algorithms in question are deterministic or non-deterministic.

Plan-Set Algorithms: Properties serve to focus on what matters for plan-set algorithms in the same way that they do for single-plan algorithms. For Top-K problems, even though bias is of limited interest, there are some interesting property bias questions to investigate. There is no reason to make the cost function a property, because we know what plan costs will be present in solutions. However, some properties of the worst plans may not be present in the sets returned by some algorithms. For diverse planning as we have defined it, the pairwise distances may be worth posing as properties. If b is the distance threshold, for any two plans r, s we know $\delta(r, s) \geq b$. Is it of interest if plans with larger separations are present? Does this bias matter? Is bias acceptable to reduce the number of the 'worst' plans returned by Top-Quality? It is likely to be more interesting to know if other properties of plans, including their costs, may not be present in the sets returned by some algorithms. Corollary 1 also suggests ways to combine property sets and find ones that are fine-grained enough to be of interest, while reducing bias.

Unifying Bias Investigations with Properties: The cost of a plan $F(s)$ and plan distance $\delta(r, s)$ can both be thought of as functions of properties of plans. When written this way, $F(\Phi(s))$ and plan distance $\delta(\Phi(r), \Phi(s))$, we can then think about how the full set of properties is 'partitioned' by the cost and distance metrics:

- Φ_F is the set of properties used as input to the cost function.
- Φ_δ is the set of properties used as input to the plan distance metric.
- Φ^c_F is the set of properties *correlated* with those properties used as inputs to the cost function.
- Φ^c_δ is the set of properties *correlated* with those properties used as input to the plan distance metric.
- Φ_r is the 'residual' properties, namely $\Phi \setminus (\Phi_F \cup \Phi_\delta \cup \Phi^c_F \cup \Phi^c_\delta)$

This strategy of using properties as the foundation for costs and distance metrics allows users to focus on which of the sets bias might, or might not, appear in. In diverse planning, we would expect bias to occur in properties used in the distance metric. Bias in the 'residuals' may not be of interest, but investigating bias in the 'residuals' may reveal unexpected correlations with properties influencing the plan distance.

7.4 Bias is Everywhere

Our theoretical foundation has focused on sets of plans, and algorithms that produce plans. But an optimal planning problem is simply one form of constrained optimization problem. *All* of our definitions and theorems apply to *any* constrained optimization problem for which a user may want many solutions. This covers scheduling, constraint satisfaction, combinatorial design problems, machine learning...*everything.*

7.5 Related Work

The survey of bias mitigation in machine learning (Hort et al. 2023) includes an extensive discussion of classifier-centric fairness metrics. Metrics may be based on the data, on the predicted outcomes, on the structure of groups (classes) the classifiers take as input, and so on. Some metrics are based on a similar notion of differences in probability of predictions. No metric appears to use a notion of the set of possible classifiers that could be produced given a set of data, which is the comparable method to that we propose.

Top-K planning has its roots in shortest path problems (Aljazzar and Leue 2011). Top-K and Top-quality planning have been the subject of considerable recent work, including (Katz, Sohrabi, and Udrea 2020) and (Katz et al. 2018). Work on eliminating some plans from consideration to reduce 'pathologies' in the sets of solutions has led to problem formulations such as Loopless Top-K (von Tschammer, Matm uller, and Speck 2022) and Subset-Top-K (Katz and Sohrabi 2022).

Diverse planning also has a long history (Srivastava et al. 2007),(Nguyen et al. 2012),(Coman and Munoz-Avila 2011),(Roberts, Howe, and Ray 2014), (Sohrabi et al. 2016), (Vadlamudi and Kambhampati 2016). (Goldman and Kutur 2015) highlights pathological behavior of some plan diversity metrics and points out problems in aggregating plan diversity measures over sets of plans, and also provides theoretical and practical solutions / new diversity metrics.

The concept of properties as a foundation of planning is also of interest. (Lehman and Stanley 2011) showed that a search space can be collapsed into a finite space (behaviour space) to model solutions' characteristics. Their motivation behind discretising the solution space was to keep track of different solutions generated by evolutionary algorithms. (Eifler, Frank, and Hoffmann 2022) describes the use of properties as a means of eliciting user preferences.

References

Aljazzar, H.; and Leue, S. 2011. K*: A heuristic search algorithm for finding the k shortest paths. 175(18): 2129 – 2154.

Chan, S.; Diakonikolas, I.; Valiant, G.; and Valiant, P. 2014. Optimal algorithms for testing closeness of discrete distributions. In *Proceedings of the* 25th *annual ACM-SIAM symposium on Discrete algorithms (SODA*, 1193 – 1203.

Coman, A.; and Munoz-Avila, H. 2011. Generating diverse plans using quantitative and qualitative plan distance metrics. In *Proceedings of the* 25th *National Conference on Artificial Intelligence*, 946 – 951.

Diakonikolas, I.; and Kane, D. 2016. A New Approach for Testing Properties of Discrete Distributions. In *Proceedings of the IEEE* 57th *Annual Symposium on Foundations of Computer Science (FOCS)*, 685 – 694.

Eifler, R.; Frank, J.; and Hoffmann, J. 2022. Explaining Soft-Goal Conflicts through Constraint Relaxations. In *Proceedings of the* 31st *International Joint Conference on Artificial Intelligence*, 4621 – 4627.

Goldman, R.; and Kutur, U. 2015. Measuring Plan Diversity: Pathologies in Existing Approaches and A New Plan Distance Metric. In *Proceedings of the* 29th *National Conference on Artificial Intelligence*, 3275 – 3282.

Hort, M.; Chen, Z.; Zhang, J. M.; Harman, M.; and Sarro, F. 2023. Bias Mitigation for Machine Learning Classifiers: A Comprehensive Survey. *ACM J. Responsib. Comput.* Just Accepted.

Katz, M.; and Sohrabi, S. 2022. Who Needs These Operators Anyway: Top Quality Planning with Operator Subset Criteria. In *Proceedings of the* 32nd *International Conference on Automated Planning and Scheduling*, 179 – 183.

Katz, M.; Sohrabi, S.; Udrea, O.; ; and Winterer, D. 2018. A Novel Iterative Approach to Top-k Planning. In *Proceedings of the* 28th *International Conference on Automated Planning and Scheduling*, 132 – 140.

Katz, M.; Sohrabi, S.; and Udrea, O. 2020. Top-Quality Planning: Finding Practically Useful Sets of Best Plans. In *Proceedings of the* 34th *National Conference on Artificial Intelligence*, 9900 – 9907.

Lehman, J.; and Stanley, K. O. 2011. Abandoning objectives: Evolution through the search for novelty alone. 19(2): 189 – 223.

Nguyen, T. A.; Do, M.; Gerevini, A. E.; Serina, I.; Srivastava, B.; and Kambhampati, S. 2012. Generating diverse plans to handle unknown and partially known user preferences. 190: 1 – 31.

Paredes, A. 2023. Structural Bias in Heuristic Search (Student Abstract). In *Proceedings of the* 16th *Symposium on Combinatorial Search (SOCS)*, 196 – 197.

Roberts, M.; Howe, A.; and Ray, I. 2014. Evaluating diversity in classical planning. In *Proceedings of the* 24th *International Conference on Automated Planning and Scheduling*, 253 – 261.

Sohrabi, S.; Riabov, A. V.; Udrea, O.; and Hassanzadeh, O. 2016. Finding Diverse High-Quality Plans for Hypothesis Generation. In *Proceedings of the* 22nd *European Conference on 390 Artificial Intelligence (ECAI 2016)*, 1581 – 1582.

Srivastava, B.; Nguyen, T. A.; Gerevini, A.; Kambhampati, S.; Do, M. B.; and Serina, I. 2007. Domain Independent Approaches for Finding Diverse Plans. In *Proceedings of the* 20th *International Joint Conference on Artificial Intelligence*, 2016 – 2022.

Vadlamudi, S. G.; and Kambhampati, S. 2016. A combinatorial search perspective on diverse solution generation. In *Proceedings of the* 30th *National Conference on Artificial Intelligence*, 776 – 783.

von Tschammer, J.; Matm uller, R.; and Speck, D. 2022. Loopless Top-K Planning. In *Proceedings of the* 32nd *International Conference on Automated Planning and Scheduling*, 9900 – 9907.