A Tale of t Metrics

Mark Roberts and **Adele E. Howe** and **Indrajit Ray**
Computer Science Dept., Colorado State University
Fort Collins, CO 80524, USA
email: {mroberts,howe,indrajit}@cs.colostate.edu

Abstract

Classical planning systems tend to focus on producing one best solution according to a single objective function or metric. In domains with metrics that can be linearly combined, action costs are a suitable choice for modeling the metric(s). In some domains, however, metrics can interact in subtle ways that may be inappropriate for a weighted objective function. Existing benchmark domains explore a small subset of potential metric interactions, typically examining one or two non-temporal metrics that rarely interact within the same action. We create a synthetic domain that allows us to systematically vary two non-temporal metrics with respect to each other and to plan length. From this domain, we make observations about the search behavior of bounded suboptimal search. We find that search performance (a) is the poorest when the two metrics are either collinear or uncorrelated regardless of their strength of correlation with plan length; (b) is best when the metrics correlate with each other in a simple curvilinear fashion that is strongly correlated with plan length; (c) degrades for curvilinear functions as the functions become weakly correlated with plan length, and (d) degrades as the metrics interact with each other or plan length in subtle or uncorrelated ways.

Domains with multiple, competing objectives (e.g., minimizing time versus money) are common but have been simplified in planning research to avoid explicitly reasoning about the trade-offs. The net-benefit track in the 2006 planning competition (Gerevini and Long 2005) and the focus on plan quality in recent International Planning Competitions (IPCs) using action costs have taken steps in this direction, capturing the trade-offs via a weighted evaluation function.

Temporal and resource reasoning lies at the heart of many significant problems in planning research. A wealth of literature exists for producing (diverse) temporal plans using a single plan metric (e.g., SAPA (Do and Kambhampati 2003), LPG-td (Gerevini, Saetti, and Serina 2006), COLIN (Coles et al. 2012)). In most cases, these temporal+metric planners embed deep reasoning to solve specific resource and time constraints that lie at the junction of planning and scheduling. It is difficult to assess the contribution of the non-temporal metrics when they are combined with temporal concerns in a weighted objective function.

We focus on numeric-only metric domains rather than examining temporal+metric planning. Our approach is to isolate the *non-temporal* metrics with the hope of assessing search behavior for these metrics. The findings we present complement the temporal+metric planning literature by probing deeper into the non-temporal metrics. We discuss how metrics are used in existing IPC benchmarks and show that these domains rarely explore interactions between the metrics *within the same action*. We then highlight some unique qualities of the domain that motivates our study: cybersecurity for the personal home computer user. The security domain does not require deep temporal or resource reasoning, but it does have multiple non-temporal metrics by which plans are evaluated.

To study metric interaction in isolation, we create a synthetic domain that allows us to vary the interactions of two metrics, x and y, with each other and with plan length. The interactions include uniform random plus three structured interactions (and their "mirrored inverses"): linear, sigmoidal, and polynomial. We examine search cost and the ability to find minimal solutions for A^*_ϵ (Pearl and Kim 1982), as implemented in MetricFF (Hoffmann 2003). We find that this implementation of A^*_ϵ has the most difficulty finding minimal solutions for collinear and random interactions, while it is most successful for simple curvilinear interactions that are strongly correlated with plan length. Uniformly scaling the metrics so that they less strongly correlate with plan length decreases the ability of A^*_ϵ to find minimal solutions. Weighting one metric more heavily than the other degrades search performance as well. These findings, though limited, suggest directions for future work in characterizing search behavior when metrics interact.

Multi-metric Domains

Our work is motivated by a non-temporal domain: identifying potential breaches for a personal computer system (Roberts et al. 2011). We are extending this domain to use four metrics not readily combined into a single objective function. The *likelihood of attack* and *cost of attack* characterize the risk associated with doing nothing about a potential threat, while *utility to the user* and *cost of intervening* characterize the reward of better securing the system while minimizing user irritation. Searching for breaches in the security domain is unique in that a property called *monotonicity* (Ammann, Wijesekera, and Kaushik 2002) eliminates cy-

cles and bounds the length of plans, but the interacting metrics create a challenge for search because they may interact in subtle ways. For example, phishing attacks have a high privacy violation cost because a novice user may provide financial or personal details to a malicious third party, thus subjecting themselves to unpleasant consequences. Since such attacks can only happen through email, a low cost intervention might be to disallow email. However, this is unlikely to be a desirable intervention from the user's perspective given the ubiquity of email use for communication. A higher-cost intervention could be educating the user, but failing to do this well may increase the risk and/or cause the user to be more confident in their ability than they actually are. The agent of this security application must reason about a set of alternative plans which balance the complex trade-offs of how likely an attack will lead to a compromised system, of the costs associated with intervening against such attacks, and of the repair costs for a (potentially) compromised system. In short, there is no single, best solution.

Existing multi-metric benchmark domains provide some inspiration in how to approach the multi-metric security domain. We highlight six non-temporal benchmark domains from IPC-2002 (Long and Fox 2003) that are represented in PDDL 2.1, Level 2 (Fox and Long 2003): Depots, DriverLog, Satellite, Settlers, Rovers, ZenoTravel. These domains were a significant advancement toward metric planning and include at least two unique metrics in addition to plan length. Radzi (2011) shows that the single, weighted-sum metric in most of these domains interacts with plan length in ways that make the problem *straightforward* to solve[1]. Only Settlers has metrics that interact with plan length in an interesting way that is not correlated with plan length. Radzi further shows that, for most domains, the metrics interact within a set of repeatable action sequences (e.g., in satellite, turning the camera to take an image) or the metrics interact within a producer/consumer model where one action increases the availability of a resource that is later consumed by another action. While these benchmarks have a variety of metrics that do interact in constrained ways, the metrics rarely contradict plan length or interact *within* the same action.

In other multi-metric applications, the metrics may have competing or subtle interactions that occur within an action. The security application is one example. Another example is an autonomous system that needs to balance the overall objectives of a mission while assessing metrics such as power consumption, remaining time, risk and safety, level of required coordination, communications delay, etc. Yet another example is a human travel agent using a mixed-initiative planning system to book an itinerary that balances cost, total travel time, airport preferences, choice of seating, etc.

[1]More specifically, Radzi (2011) distinguishes metric straightforwardness into a spectrum of strictly straightforward (plan length is a proxy for the metric), straightforward (plan length may be a proxy), semi-straightforward (plan length and the metric may diverge but the metric is monotonic in the plan length), and expressive (the metric may contradict plan length). Radzi examines several new domains for the semi-straightforward category and leaves to future work the set of expressive domains, of which we believe the security domain may be one case.

Controlling for Metric Interactions with a Synthetic Domain

To study the impact of metric interactions on search in a controlled way, we create a synthetic domain. All solutions in the domain must have the same plan length and the metrics must be systematically varied across the operators of the domain. Figure 1 presents a pictorial view of the synthetic domain, which is essentially a fully connected planning graph. Nodes in this graph are states and arcs are transitions (operators). An $m \times n$ graph has m layers with each layer containing n states. In planning terms, m determines the plan length of any valid solution, while n determines the granularity of the metrics as discussed below.

States are labeled $s_{ij} \in \mathcal{S}$, where $i = \{0, 1, .., m\}$ indicates the layer, and $j = \{0, 1, .., (n-1)\}$ indicates one of the states within a layer. The initial ($i = 0$) and goal ($i = m$) layers have only one state; these are respectively labeled 'Init' and 'Goal' in Figure 1.

Transitions between states are ordered pairs $(s_{ij}, s_{i'j'}), i < i', j < j'$ and labeled $a_{(ij,i'j')} \in \mathcal{A}$. The middle layers of the graph are fully connected with the next layer, so there is a path from any state in level i to all states at the next level ($i' = i + 1, j' = \{0, 1, .., (n-1)\}$). There are $(m-2)n^2 + 2m$ transitions. Each arc in the graph represents a single operator, where the precondition links to a state at level i and the postcondition links to a state at level $i + 1$. This version of the synthetic domain is lacking some key features of many applications: there are no negative effects and no cycles[2]. We plan to add these features into future studies of the synthetic domain.

Adding Metrics To The Synthetic Domain

Table 1 shows the weights of the transition matrix for the 3×3 problem. The weight of an arc is set to the value of the node it leads into (shown as a small integer at the top of the states in Figure 1). The arcs leading into the left-most states ($a_{i0} \forall i$) are all set to 0. Arcs leading into the right-most states ($s_{i(n-1)} \forall i$) are set to 1000. Arcs in the middle interpolate the distance between 0 and 1000. Formally, the values depend on the jth column: $a_{ij} = j * 1000/(n-1)$, where $\forall ij, 0 < i < m, 0 \leq j < n$. Thus, the number of states in each layer, n, determines the granularity of the metrics across the graph.

The three metrics x, y and z are functions of the transition matrix; we only set the value of x or y if a transition exists. The ranges of all metrics are integers $[1, 1000]$ to ensure some numeric effect in every action. Values of x are obtained: $x_{(ij,i'j')} = max(1, w_{(ij,i'j')})$. For brevity, we drop the subscripts and say $x = max(1, w)$. To obtain y, we apply functions that control the interaction between x and y. These functions are listed in Figure 2 and plotted in Figure 4. The random (ran) function is a no-interaction baseline. The other functions vary the interaction with functions (and their "mirrored inverses"): linear, lin (nil), sigmoidal, sig (gis), and polynomial, pol (lop). To aid in reading, note that the letters are reversed for each inverse function. The

[2]The monotonicity property (Ammann, Wijesekera, and Kaushik 2002) of the security domain motivates this restriction

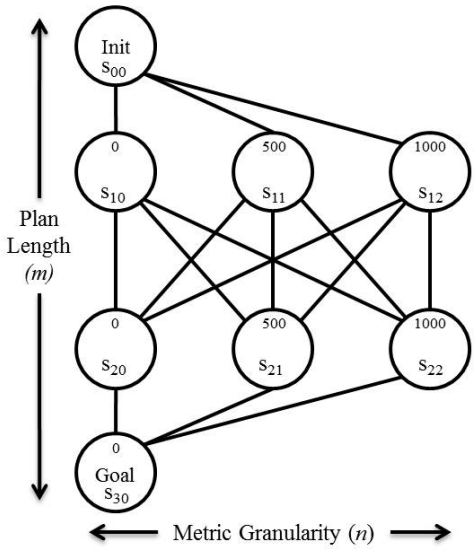

Figure 1: A graph of the 3×3 synthetic domain. Transition weights are at the top of the node an arc leads into.

$$x = max(1, w) + \epsilon$$

$$ran(w) = c \cdot uniform(1, 1000) + \epsilon$$

$$lin(w) = x = max(1, w) + \epsilon$$
$$nil(w) = max(1, 1000 - w) + \epsilon$$
$$sig(w) = round(1000 * (1/(1 + e^{-(w-500)/110}))) + \epsilon$$

$$gis(w) = round(1000 * (1/(1 + e^{(w-500)/110}))) + \epsilon$$
$$pol(w) = round(1050 * (1/exp(w/10)^{0.05})) + \epsilon$$
$$lop(w) = round(1000 * (1000^3 - w^3/1000^3)) + \epsilon$$

Figure 2: The functions used to vary metric interaction. See Figure 4 for a plot of each function.

polynomial function is rotated to provide a more interesting optimization metric. The fractional component of any function is truncated, and random noise $\epsilon = round(\mathcal{N}(0, 10))$ is added to all functions to generate randomized problems. If the function or the random noise cause a negative value, it is set to 1.

We further partition the functions into two sets. The "easy" functions $E = \{ran, lin, nil, sig\}$ show no real trade off between x and y or have many minimal solutions (as in nil). The "difficult" functions $D = \{gis, pol, lop\}$ exhibit a trade-off with a small number of minimal solutions, although multiple symmetric solutions may exist because we discretize the functions. The ordering of the functions in Figure 2 and in later plots maintains the relative order of the easy and difficult partitions.

We apply the transition matrix and the values of x and y to create PDDL problems for a specific $m \times n$ size; this

	1,0	1,1	1,2	2,0	2,1	2,2	3,0
0,0	0	500	1000				
1,0				0	500	1000	
1,1				0	500	1000	
1,2				0	500	1000	
2,0							0
2,1							500
2,2							1000

Table 1: The transition matrix for 3×3.

```
(define (domain synthetic-4-3-nil)
  (:requirements :equality :typing :fluents)
  (:types State)  (:predicates (state-active ?s - State))
  (:functions (x) (y) )
  (:action Apply-00-10  :parameters (?state-00 - State)
   :precondition (and (state-active ?state-00)
                      (= ?state-00 State-00) )
   :effect (and (state-active State-10 )
                (increase (x) 1)
                (increase (y) 1000) ) )
   ...)

(define (problem synthetic-4-3-nil-x)
  (:domain synthetic-4-3-nil)
  (:objects State-00 - State
            ...
            State-40 - State)
  (:init (state-active State-00) (= (x) 0) (= (y) 0))
  (:goal (state-active State-40))
  (:metric minimize (x) ) )
```

Figure 3: Partial PDDL for the domain and problem $4 \times 3^x_{nil}$.

yields one domain file and 21 problem variants (i.e., seven functions, three metrics). Metrics are applied in the operator effects with `(increase (f) value)`. The problems are labeled $m \times n^{\text{metric}}_{\text{function}}$, where the function is one of $\{ran, lin, nil, sig, gis, pol, lop\}$ and the metric is one of $\{x, y, z\}$. Figure 3 shows the PDDL for $4 \times 3^x_{nil}$.

Method

We used the A^*_ϵ algorithm as implemented in the newest version of MetricFF (Hoffmann 2003), a planner that directly supports the full complement of PDDL 2.1, Level 2. Instead of popping the best solution ($s = min(q)$) from the top of the priority queue at each iteration, A^*_ϵ (Pearl and Kim 1982) selects a solution from the top K solutions (i.e., a focus list) on the top of the queue. A solution s' is in K if $f(s') <= 5f(s)$ and $h(s') = h(s)$. We only use MetricFF with cost optimization enabled. This implementation does not use helpful actions nor the initial enforced hill climbing stage. We use $m \times n = 15 \times 29$ because it was as large as MetricFF could still handle.

We show box-and-whisker plots of CPU time to a completed plan. For plan cost, A^*_ϵ can produce suboptimal so-

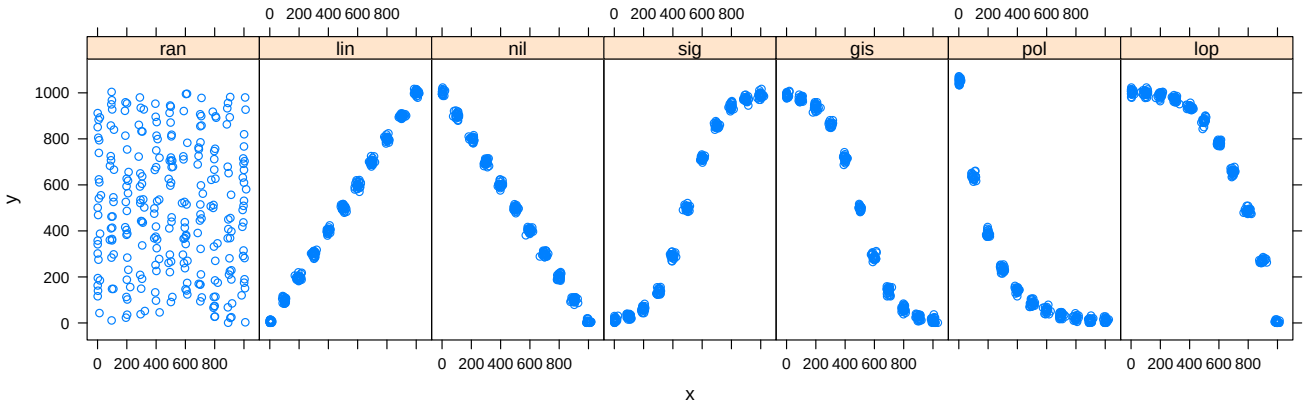

Figure 4: Examples of the interaction between metrics x and y.

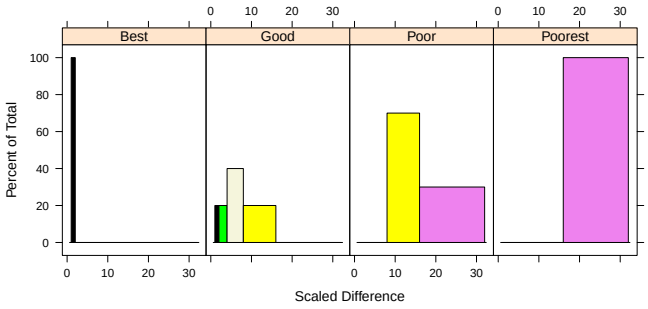

Figure 5: An example histogram explaining how to read later plots. Thinner bars to the left are best. Wider bars to the right indicate poorer performance. The y-axis is the percent of total problems (out of 30) that fall into each bin.

lutions. So we state A^*_ϵ ϵ-minimizes a function if most of its solutions fall within 8 times the optimal. We plan to explore more deeply the weight value within A^*_ϵ to determine the best setting. To plot how well A^*_ϵ minimizes the plan, each plan π is first scaled, $cost(\pi)/cost*$, where $cost*$ is the minimal solution found using a simple dynamic programming algorithm. A minimal plan receives a 1; other solutions are factors of how much worse they are from the minimal solution. We then histogram these scaled values for the 30 problems from each combination of function and metric using bin sizes of $1, 2, 4, 8, 16,$ and $>= 32$. Figure 5 shows a comparative example of what we want to see for optimal and worsening solution quality. Thicker bars indicate higher variance, while the right-most bars indicate the poorest relative performance. Note that the y-axis in these plots is the percent of total problems, which for each function/metric combination is 30 problems.

All experiments are run on a 2.7Ghz Quad Core i7 with a time limit of 15 minutes and memory limit of 1GB.

The Impact of Metric Interaction

Many planners are designed to minimize a single objective function. So we expect to observe little performance difference when minimizing either x or y alone.

Hypothesis 1: *For all functions, (a) A^*_ϵ will ϵ-minimize both x and y (b) with insignificant differences in CPU time between the two metrics.*

Table 2 (top) – this and all subsequent tables and figures are at the end of the paper – shows the runtimes of x and y along with the p-values for a pairwise t-test between x and y. These are also plotted on a log scale in Figure 6 (top). All the runtimes between the two metrics were significant ($p < 0.0073$) except the linear functions[3]. Figure 7 (left) shows a histogram of the scaled differences for how well A^*_ϵ minimizes x and y. A^*_ϵ has trouble finding the ϵ-minimal solutions.

There is a marked difference between minimizing x and minimizing y for some problems. Both metrics have a range that is represented equally in the actions at each layer of the graph. However, x is sampled at specific points that are interpolated between $[1, 1000]$ while y is sampled uniformly randomly in that same range. This sampling bias is evidenced in the vertical 'bands' for x seen in Figure 4 that are absent for y. This leads to more potential values for y than x, which appears to be more challenging for search. We plan to confirm this explanation in future work.

So we can conclude that, for the most part, hypothesis 1(a), is not validated as ϵ-minimal solutions are not found. We also conclude that hypothesis 1(b) is not supported, as there are significant differences in CPU time between minimizing x and y.

Comparing Easy and Difficult Functions

For minimizing z, we expect to see quality and performance degrade as the xy trade off becomes more challenging. In

[3]The Bonferroni adjustment controls the experiment-wise error of 7 pairwise comparisons at $\alpha = 0.05$, so the critical value for p is $0.0073 = 1 - (1 - 0.05^{1/7})$.

the following two hypotheses, we examine planner performance on the easy and difficult metric interactions. The intuition behind this hypothesis is that collinear functions are easy but non-collinear functions are difficult. So, finding the minimum will take longer and happen less frequently under difficult interactions.

Hypotheses 2: *For the easy functions (ran, lin, nil, sig), (a) A^*_ϵ will ϵ-minimize z and (b) with insignificant differences between the CPU time of minimizing x, y, and z.*

Figure 7 (left) shows that A^*_ϵ successfully finds ϵ-minimal solutions in z for nil, gis, pol, lop. In terms of runtime, Figure 6 (top) shows there is often a significant cost (10 to 100 times more) to minimizing z except in the simplest collinear functions lin and sig. The runtime for minimizing z is significantly different except for lin.

Hypothesis 3: *For the difficult functions (gis,pol,lop), (a) A^*_ϵ will not find the global minimums in z, and (b) the runtime will be significantly different between D and E. Further, (c) the runtime will be insignificant between pol and lop, but (d) will be significantly different between gis and $\{pol, lop\}$.*

Figure 6 (top) shows that the runtimes between the E and D problems are distinct, with the D problems usually taking more runtime. The runtimes between pol and lop, while overlapping, are statistically different; a Tukey HSD test run on the runtimes of both the E and D does not group any of the functions together. Figure 7 shows that A^*_ϵ successfully finds the minimal solutions for gis, pol, lop.

So we conclude that CPU time increases for the difficult functions but there is not otherwise similar performance with the easy/difficult groupings. A^*_ϵ finds minimal solutions for all three functions.

The Impact of Plan-length Correlation (PLC)

Due to using the planning graph as a heuristic function, many planners may implicitly rely on plan length to guide search. If this is true then we should expect to observe significant differences between the solutions and the runtime behavior of solving plan length correlated problems versus uncorrelated problems.

Unfortunately, we cannot completely test these intuitions with the current version of MetricFF, which only supports actions that move the metric in the opposite direction from the objective (i.e., when minimizing and given a $value$ and function f for the action, we can only use `(increase (f) value)` and cannot use `(decrease (f) value)` or `(increase (f) -value)`. This limits the kind of evidence we can collect for the behavior of best first search w.r.t. controlling PLC. So we discuss two additional experiments that assess how sensitive A^*_ϵ is to scaling x or y.

Recent research shows that cost-based search is sensitive to the ratio of the operator costs (i.e., (Wilt and Ruml 2011), (Cushing, Benton, and Kambhampati 2011)). Sroka and Long (2012) assess the metric sensitivity of planners and show that MetricFF (and other planners) can generate more diverse solutions by varying the constraintedness of resources in a logistics domain. We are interested in how the search behavior changes when metric interactions vary in addition to the scaling.

Uniformly Scaling x and y

We create a set of under-correlated problems that simply scale the original metric values in actions by 0.3. Figure 6 (bottom) and Table 2 show the runtime distributions for the under-correlated problems. The results parallel those of the correlated problems except for a more significant difference in gis, which can be explained by an outlier present in the original runs.

Figure 7 (right) shows histograms of the scaled difference from minimal solutions for the under-correlated problems. When minimizing x or y, A^*_ϵ finds equal or better solutions for the under-correlated problems, except for ran. In contrast, when minimizing z, A^*_ϵ finds worse solutions except for lin and sig.

Scaling either x or y in z

Another way we can control for PLC is to scale either x or y when minimizing z. This leads to a skewed evaluation function that favors one axis over another. We introduce functions that vary the weight of x and y (we only show the functions for x, but y is similar).

$$z_{2x} = 2x + y$$
$$z_{5x} = 5x + y$$
$$z_{10x} = 10x + y$$
$$z_{25x} = 25x + y$$
$$z_{50x} = 50x + y$$

Figure 8 shows the demonstrates that solutions tend to get worse as the scale increases (to the right). This trend is less pronounced (or absent) in the random and collinear functions (lin and sig) while very evident in the remaining functions. We do not show the plots for scaling y, but the trend is the same.

Summary and Future Work

We examine a synthetic domain that allows us to vary the interaction of two metrics, x and y, in a weighted objective function, z, while partially controlling for plan-length correlation. One of the more surprising findings is that A^*_ϵ search performs quite poorly when minimizing collinear functions ($y = x$ and $y = sigmoid(x)$), which suggests that researchers should avoid combining x and y when they are (nearly) collinear. A^*_ϵ works well for curvilinear functions such as polynomials and an "inverted" sigmoid. However, we discovered that scaling the metrics dramatically reduced search effectiveness. Poorer performance occurred when the metrics were uniformly scaled to control, at least in part, for plan-length correlation. Search performance also degraded as one metric was weighted more heavily.

There are limitations of the work that should be addressed in future work. We started with easier bi-criteria functions (linear and curvilinear) and plan to extend to more complex interactions and more metrics. We also need to control for

the discretization of x that led to better results than y. Uniformly selecting x should solve this problem. We identified at least some evidence for plan-length correlation and intend to explore this further. Our choice of MetricFF, A^*_ϵ and metric planning limits the findings to a single approach; we need to generalize our results to preference and cost-based planners as well as other metric planners. It is unclear how much the weight of A^*_ϵ or the heuristic accuracy could impact the search results, which needs to be addressed with a deeper study of the parameters for A^*_ϵ. It may also be fruitful to compare the plans found with worst cost plans rather than the optimal cost.

Ultimately we are interested in combining this work with search for diverse alternatives plans as in the work of (Nguyen et al. 2012; Coman and Munoz-Avila 2011; Roberts et al. 2012; Khouadjia et al. 2013; Radzi 2011; Sroka and Long 2012). In addition to examining non-temporal metrics, we want to focus on generating multiple alternative plans rather than a single plan. We intend to extend our set of domains to those found in Radzi (2011) and Sroka (2012) . We also hope to extend the findings of this paper to the security domain that motivates our initial interest in this topic.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 0905232. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. We also thank the reviewers of this paper for their insightful comments that helped improve the paper.

References

Ammann, P.; Wijesekera, D.; and Kaushik, S. 2002. Scalable, graph-based network vulnerability analysis. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, 217–224.

Coles, A. J.; Coles, A. I.; Fox, M.; and Long, D. 2012. Colin: Planning with continuous linear numeric change. *Journal of Artificial Intelligence Research* 44:1–96.

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

Cushing, W.; Benton, J.; and Kambhampati, S. 2011. Cost based satisficing search considered harmful. In *Working notes of the Workshop on Heuristics for Domain-independent Planning at ICAPS-2011*.

Do, M. B., and Kambhampati, S. 2003. SAPA: A multi-objective metric temporal planner. *Journal of Artificial Intelligence Research* 20:155–194.

Fox, M., and Long, D. 2003. PDDL2.1: An extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research* 20:61–124.

Gerevini, A., and Long, D. 2005. Plan constraints and preferences in PDDL3. Technical Report RT 2005-08-47, Dept. of Electronics for Automation, University of Brescia, Italy.

Gerevini, A.; Saetti, A.; and Serina, I. 2006. An approach to temporal planning and scheduling in domains with predicatable exogenous events. *Journal of Artificial Intelligence Research* 25:187–231.

Hoffmann, J. 2003. The Metric-FF planning system: Translating "ignoring delete lists" to numeric state variables. *Journal of Artificial Intelligence Research* 20:291–341.

Khouadjia, M.-R.; Schoenauer, M.; Vidal, V.; Dréo, J.; and Savéant, P. 2013. Multi-objective AI planning: Evaluating DAEyahsp on a tunable benchmark. In *Proceedings of the 7th International Conference on Evolutionary Multi-Criterion Optimization (EMO-2013)*, LNCS. Sheffield, UK: Springer.

Long, D., and Fox, M. 2003. The 3rd international planning competition: Results and analysis. *Journal of Artificial Intelligence Research* 20:1–59.

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

Pearl, J., and Kim, J. H. 1982. Studies in semi-admissible heuristics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 4(4):392–399.

Radzi, N. H. M. 2011. *Multi-objective planning using linear programming*. Ph.D. Dissertation, The University of Strathclyde.

Roberts, M.; Howe, A.; Ray, I.; Urbanska, M.; Byrne, Z. S.; and Weidert, J. M. 2011. Personalized vulnerability analysis through automated planning. In *Workshop on Security and Artificial Intelligence (SecArt-11) in Working Notes of IJCAI-11. Barcelona, Catalonia, Spain. July 16-22*.

Roberts, M.; Howe, A. E.; Ray, I.; and Urbanska, M. 2012. Using planning for a personalized security agent. In *Workshop on Problem Solving using Classical Planners in Working Notes of the 26th AAAI Conference on Artificial Intelligence*. AAAI Press.

Sroka, M., and Long, D. 2012. Exploring metric sensitivity of planners for generation of pareto frontiers. In *Starting AI Researchers (STAIRS 2012)*, volume 241 of *Frontiers in Artificial Intelligence and Applications*, 306–317.

Wilt, C., and Ruml, W. 2011. Cost-based heuristic search is sensitive to the ratio of operator costs. In *Fourth Annual Symposium on Combinatorial Search (SoCS)*.

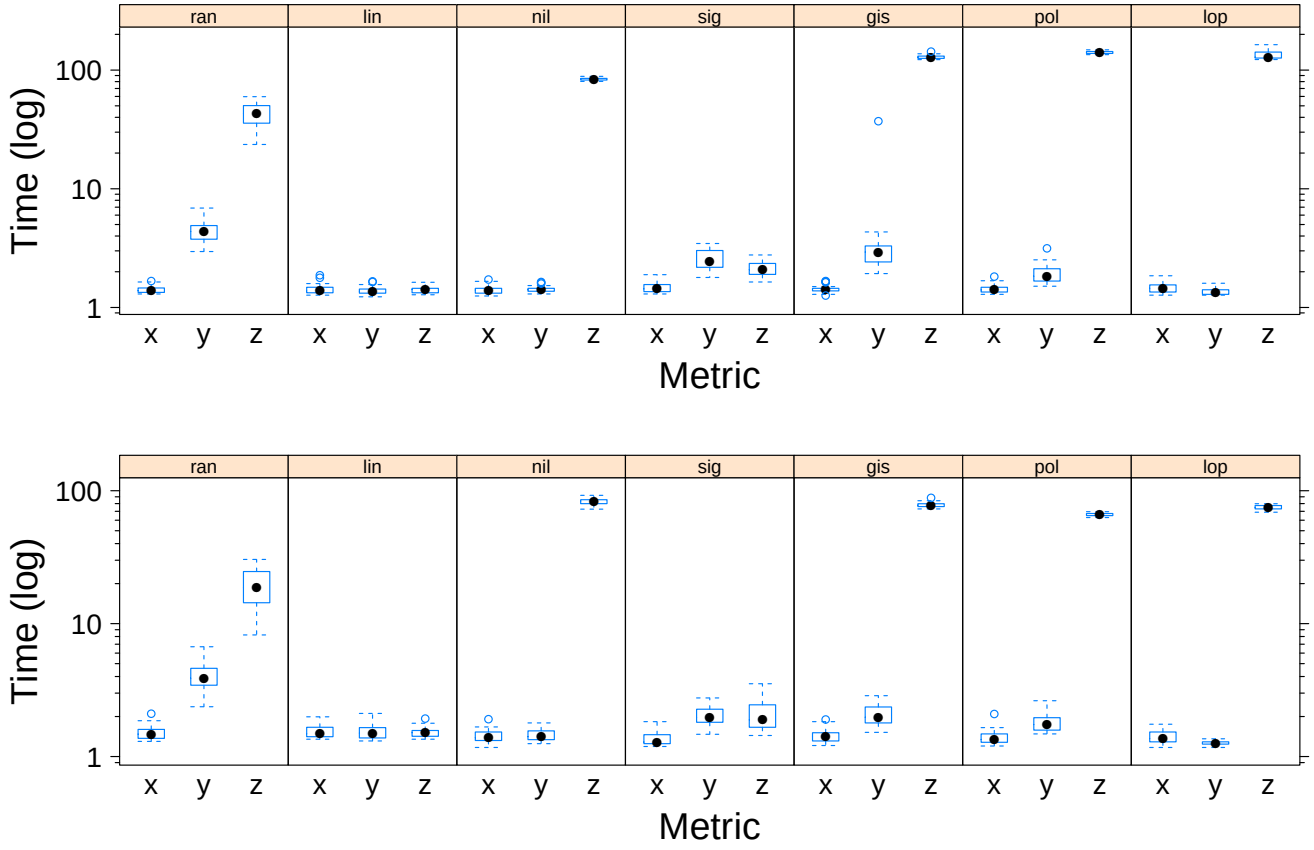

Figure 6: Runtime distributions of all functions and metrics for the original problems (top) and the under-correlated problems (bottom).

			x		y	
	Sig.	p-val	Avg. time	σ	Avg. time	σ
ran	***	0	1.41	0.101	4.48	0.958
lin		0.248	1.42	0.139	1.38	0.107
nil		0.669	1.41	0.109	1.42	0.088
sig	***	0	1.48	0.152	2.52	0.476
gis	*	0.034	1.43	0.104	3.99	6.28
pol	***	0	1.43	0.115	1.94	0.372
lop	***	0.001	1.47	0.148	1.36	0.083
ran	***	0	1.52	0.189	4.1	0.953
lin		0.984	1.54	0.169	1.54	0.203
nil		0.751	1.44	0.162	1.45	0.153
sig	***	0	1.37	0.174	2.03	0.297
gis	***	0	1.43	0.162	2.09	0.338
pol	***	0	1.4	0.185	1.82	0.332
lop	***	0	1.42	0.16	1.27	0.048

Table 2: Significance, p-value, and statistics (average time and standard deviation) for x and y on the original problems (top) and under-correlated problems (bottom).

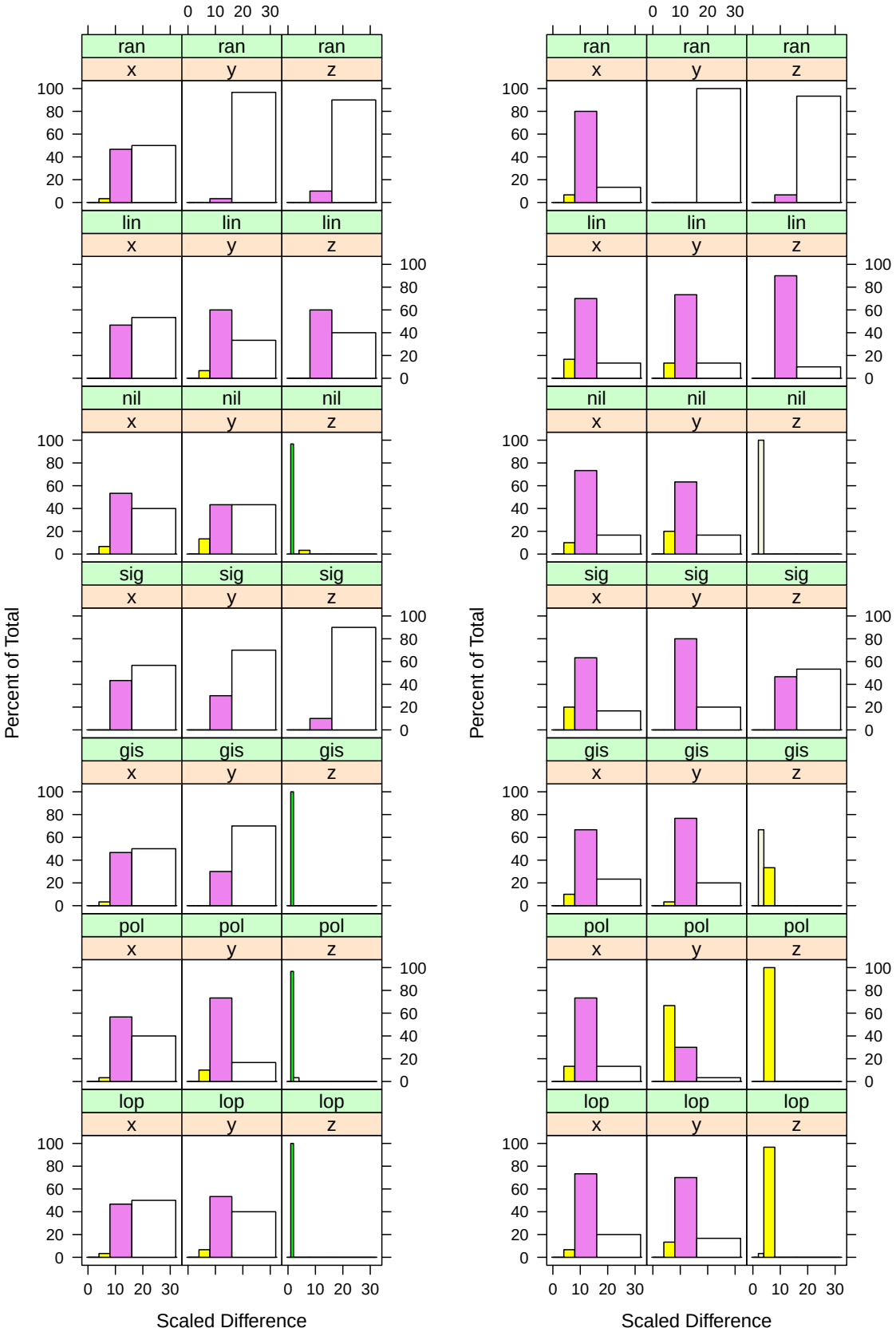

Figure 7: Log-Histograms of the scaled difference from the minimum for the original problems (left) and the under-correlated problems (right). Bins for the bottom axis are set at $\{0, 1, 2, 4, 8, 16, \geq 32\}$ to provide a visual representation of how well an algorithm does. Better performance is indicated by thinner and taller bars to the left. For more details, see the discussion of Figure 5.

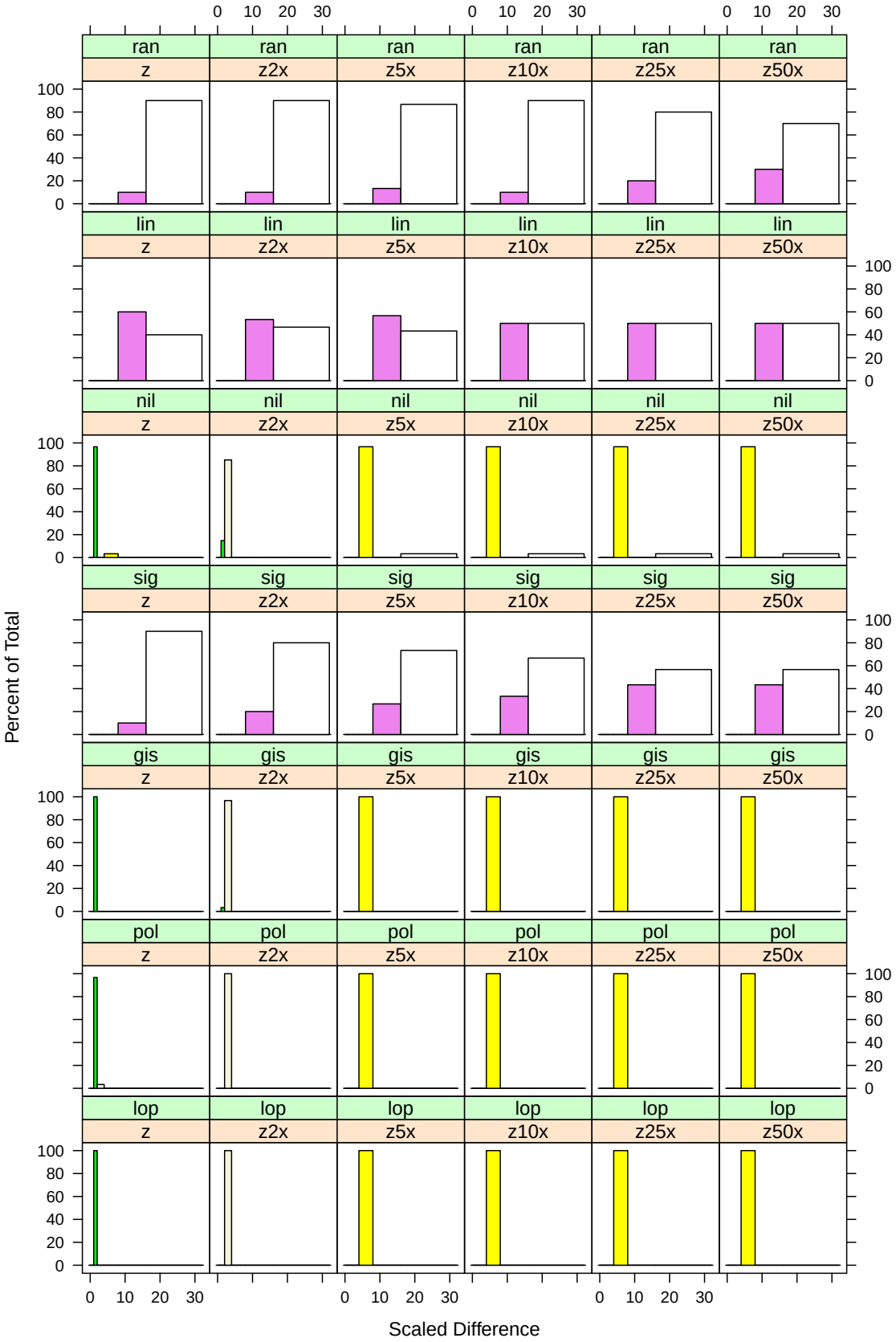

Figure 8: Log-Histograms of the scaled difference from the minimum for the z_{*x} problems. Bins for the bottom axis are set at $\{0, 1, 2, 4, 8, 16, \geq 32\}$ to provide a visual representation of how well an algorithm does. Better performance is indicated by thinner and taller bars to the left. For more details, see the discussion of Figure 5.