STAIRS 2012
K. Kersting and M. Toussaint (Eds.)

doi:10.3233/978-1-61499-096-3-306

Exploring Metric Sensitivity of Planners for Generation of Pareto Frontiers

Michal Sroka [a], Derek Long [b]

[a] *Michal.sroka@kcl.ac.uk Department of Informatics, King's College London, UK.*
[b] *Derek.long@kcl.ac.uk Department of Informatics, King's College London, UK.*

Abstract. This paper explores how current planners behave when exposed to multiple metrics, examining which of the planners are metric sensitive and which are not. For the metric insensitive planners we propose a new method of simulating metric sensitivity for the purpose of generation of diverse plans close to a pareto frontier. It is shown that metric sensitive planners are good candidates for generating sets of pareto optimal plans.

Keywords. Planning, multiple objectives, solution set, metric sensitivity

1. Introduction

The problems we face and solve in practice are usually complex and it is often insufficient to consider only one possible solution. Most of the state of the art planners nowadays focus on delivering a single solution either fast or with a high quality. In order to be more appealing to a decision maker (DM) who uses the planning system, it would be better to construct a collection of very good plans from which the DM could choose the solution to their problem.

Some of the current planning system offer a possibility to specify an objective function and then produce a plan which achieves the goal and also minimizes the objective function. Problems, however, cannot be expressed and evaluated using a single objective function because defining the whole set of constraints and objectives as a single function is extremely difficult and in some cases impossible. To do so would require the DM to specify explicitly the relationship, in the form of relative weights, between various aspects of the problem. The objective would have to balance relative costs of resources and the rewards for different ways of achieving goals.

For example a manager deciding on how to ship a set of packages from a set of locations to a different set of locations needs to evaluate the solutions in terms of cost (the cheaper the better), risk (the safer the better), time (the faster the better), employee satisfaction and so on. A typical manager will not be able to determine numerical weights between the objectives and therefore combining them into a single objective function will not yield optimal plans due to incorrect weights. We cannot simply say that our objective = cost -risk + time + employee satisfaction, because in most problems the objectives are not equally important.

One of the solutions to this problem is to find a set of pareto optimal plans well distributed across the pareto frontier and present them to the decision maker who can then easily make the trade-off between the solutions. Producing three different plans where one requires employees to carry the packages to their destination, the second has packages transported on mechanical trolleys by employees, and the third uses a hired courier to do all the job, is more informative for the decision maker. This approach helps him make an informed decision without the need of specifying weightings between objectives.

From the above example it is clear that it is very important to produce sets of plans and not a single plan. This could be achieved using planners which are sensitive to the change of objectives. These planners can be directed into different areas of a search space using weights on the objective function. Metric sensitive planners are a crucial element in generating well populated pareto frontiers of plans. It allows us to explore distinct areas of the search space and find qualitatively different, in terms of metrics, plans. This challenge is very interesting as it has not been widely explored. This paper explores the extend to which the current planners are metric sensitive, and determines how current planners respond to changes in the objectives. We also show how some of the metric insensitive planners can be used to generate distinct solutions.

2. Background

A planner is a piece of software which for a given domain and a problem description outputs a plan, or a set of plans which solves the given problem in the given domain. A planning domain is usually described using PDDL [4]. Planning problems are described using sets of propositions and numeric fluents. An example of the proposition is (on A B) which, in the context of blocks-word, means that the block A is on the block B. The planning problem usually describes a start state and the goal condition in a given domain. The planner then outputs a sequence of actions which if applied take us from the start state to a state satisfying the goal. The planning problem also contains an objective function which is the main focus of this paper. The objective functions are different from goal states because they do not need to be satisfied in order for the problem to be solved. The plan can be awarded extra value if it satisfies conditions given by objectives, or if it minimizes cost functions described in objectives.

Definition 1. *An **objective function** for a plan, ρ_i, domain D and problem P, is a function $\Theta(D,P)(\rho_i)$ which assigns a score to ρ_i. For a problem with many objective functions we will denote them as Θ_k where k= 1...N where N is the number of objective functions.*

The long term goal of this research programme is to generate a set of plans, where the plans are all good quality, in terms of the objective functions, and significantly different from each other.

By different plans we mean plans where a metric which we define later is larger than a specific threshold. A good quality plan in terms of multiple objectives is one that only few others can have better objective values. We would like to find plans such that there are no other plans which are better in terms of all of the objectives. This set of non-dominated plans is called a pareto frontier.

Definition 2. ***Plan Domination**. Plan ρ_0 dominates plan ρ_1 if there exists an objective function, Θ_i, such that $\Theta_i(\rho_0) < \Theta_i(\rho_1)$ and for all other objective functions $\Theta_j(\rho_0) \leq$*

$\Theta_j(\rho_1)$ *In other words, the dominating plan is better in at least one objective function and no worse in the rest.*

Generating a pareto frontier is a known challenge and we discuss it in Section 3.3.

We would like to be able to generate the set of non-dominated and different plans using an existing planner. It seems clear that the planner must be able to generate different plans for the same problem with different objective functions. This requires the planner to be metric sensitive.

Definition 3. *A planner is **metric sensitive** if it generates different plans in response to changes in the plan metric.*

For the purpose of this paper we treat metric sensitivity as a binary property. A planner either is or is not metric sensitive. This can be tested by using the planner to generate plans for the same domain and problem using different objective functions. If a change in the objective function causes the planner to generate different solutions, and if these solutions correlate with the changes in metrics, then the planner is metric sensitive. In future research we will aim to identify a metric to measure how metric sensitive planners are as this is an important property in terms of multi-objective planning.

One way to consider two plans to be similar (following [15]) is if they use the same actions, visit the same states or share the same causal links. Where a causal link is a tuple (a1, p, a2) where action a1 achieves proposition p which is a precondition of action a2. The distance between two plans can be defined in terms of the degree of similarity under one of these measures. Plans can then be considered qualitatively different if they are sufficiently different under this measure. However, we focus on measuring the distance between plans using only the objective functions provided and combining them in Euclidean distance which is defined later. This approach assesses plans directly on how they can trade-off one metric to the other and find plans different in terms of this metrics.

Bias towards plans which appear different to the planner, but are essentially similar to the decision maker, is addressed automatically by selecting appropriate objective functions. For example, suppose we want to transport a package from location A to location B, and we have two routes to choose, namely, via C1 and via C2, we assume that both are equal in time and fuel consumption, as in Figure 1

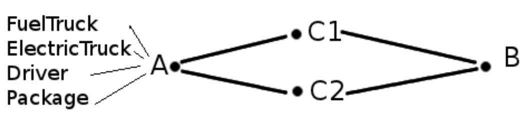

Figure 1. (at FuelTruck A) (at ElectricTruck A) (at Driver A)(at Package A) Goal: (at Package B)

We can choose between two vehicles, a diesel vehicle and an electric vehicle. Here are some example plans solving that problem:

P1	(embark driver diesel-truck), (load package diesel-truck) (drive diesel-truck A C1), (drive diesel-truck C1 B), (unload package diesel-truck)
P2	(embark driver diesel-truck), (load package diesel-truck), (drive diesel-truck A C2), (drive diesel-truck C2 B), (unload package diesel-truck)
P3	(embark driver electric-truck), (load package electric-truck), (drive electric-truck A C1), (drive electric-truck C1 B), (unload package electric-truck)

The difference between plans (1) and (2) in terms of states and actions (using grounded actions) is significant. However, in terms of an objective function: (minimize (+ time electricity-cost fuel-used)) they are both the same. If going via C1 makes a difference, and is more desirable, the decision maker should have added a statement like (* -1 (visited C1)) to the objective function, which would then favor going via C1.

Definition 4. ***Distance between plans*** *$|\rho_1\rho_2|$ is an Euclidean distance in the space described by the metrics.*

$$|\rho_1\rho_2| = \sqrt{\Sigma_{i=0}^{n}(\Theta_i(\rho_2) - \Theta_i(\rho_1))^2} \tag{1}$$

$|\rho_1\rho_2|$ is the distance between plans $\rho_1 and \rho_2$ based on it objective functions Θ_i. Both plans are evaluated using the same set of objective functions.

If we choose to use the distance measure as defined in Definition 4, with the single objective function defined above, we would find that plan (1) is the same as (2) but very different from (3). It is clear that having this definition really makes a difference since now the DM is presented with a set of plans where a visible trade-off between resources is made and, therefore, the DM can make an informed decision. Also, since the route of the trucks does not affect our objective function, the decision on the route is abstracted out from the problem for the DM. We could solve the same problem with an objective function favoring a more interesting or less dangerous route, or a route passing by a favorite book shop of the driver. This, in turn, would give an appropriate choice to the driver and would abstract the unnecessary, for the driver, decision about whether to use an electric or diesel truck.

3. Background - Related Work

3.1. Generating Sets of Plans

An approach to generating sets of plans where metric sensitivity was necessary is examined in [15] and [16]. A method of generating dDISTANTkSETs is presented, where dDISTANTkSETs are sets of k different plans where each of the plans is distant from all others by minimum distance d. Where the distance is measured in one of the 3 ways. 1) based on Actions 2) based on Causal links or 3) based on states visited. Where a causal link is a tuple <a1, p, a2> where action a1 achieves a proposition p which is a precondition of action a2. Authors approach is driven by the assumption that ‘the decision maker can not provide a full description of their expectations [15]. Therefore the planner can not produce a single plan which satisfies the decision maker, due to the lack of knowledge. The outcome of the planning process in this case should be a set of different plans (suggestions) which DM can use to make a decision. The base for this is the assumption that the DM can easily assess ready plans, but not weights between the objectives. In this research two planners are used, LPG-Numeric and GP-CSP; both exhibit a metric sensitive behaviour. The output of the process is a set of plans separated by a minimum threshold d. Integrated Convex Preference (ICP) is used to score the sets of plans. Where ICP is given as:

$$ICP(\rho^*) = \sum_{i=1}^{k} \int_{w_{i-1}}^{w_i} h(w)(w \times t_{p_i} + (1-w) \times c_{p_i})dw \quad (2)$$

For each plan from the set ρ^* a weighted sum of time t_{p_i} and cost c_{p_i} of that plan is calculated. $w_0 = 0$ and $w_k = 1$. For more details please refer to [15] Section 3.2.

The ICP score has been integrated into LPG and was used to drive it towards plans different from already found. Using ICP in that form forces the planner to find multiple different solutions.

However this does not address the issue of generating different pareto optimal solutions in terms of objective functions. The stress is on diversity, the quality of some of the solutions suffers, as demonstrated in Section 4.2.1 below. This is due to the fact that each of the solutions in the set does not have to be optimal and, even though the planner will automatically try to find best solutions, it will try to maximize diversity, the ICP function, of the set and not the total quality of the set. When the good solutions which it can find are all very close to each other, with regards to the metrics used, the planner generates other solutions different to those found already but with poorer values under the objective function.

According to the ICP metric a time is traded with cost, where time is the time it takes to execute the plan and the cost is a cost of the plan. The cost of the plan represents combined objective functions and the plan execution time is a metric which all planners try to minimize automatically therefore this approach is similar to planning with one objective function (cost), because a state of the art planners always try to minimize the time span of the plan. Result of the paper is how a planner (LPG-metric in that case) can trade-off plan length which is equivalent to the time and the plan cost, which is equivalent to the aggregated objective function. In this paper we present a method of planning without the need of explicitly specifying one objective function (cost) a-priori which allows us to specify multiple objective functions and planner will automatically present a trade-off between them.

3.2. Objective Functions

Objective functions were introduced into PDDL[4] from version 2.1 onwards. An example of a metric specification is

(:metric minimize (+ (* 2 (fuel-used car)) (fuel-used truck))).

The idea of objectives was developed further in PDDL3 [6] where more sophisticated ways of specifying preferences are described. A broader syntax from modal temporal logic is introduced. Preferences on plan trajectory can be expressed using a combination of the following: (always f), (sometime f), (at-most-once f), (sometime after f g), (sometime-before f g).

3.3. Generating a Pareto Frontier

Mathematical methods of calculating the pareto frontier are described in many papers, for example [18], [10], [12] and [17]. In these the authors explore various approaches to calculating the pareto frontier. Although those methods cannot be directly applied to planning, we can benefit a lot by understanding them as many of the approaches can be translated into the planning context. The methods are divided into three main categories, depending on when the knowledge about DM preferences is known. These categories are: a priori articulation of preferences, a posteriori articulation of preferences, no articulation of preferences. Methods surveyed in [12] include various weighting approaches, lexicographic method, bounded objective function, goal programming, physical programming (PP), normal boundary intersection (NBI), normal constraint method (NC), genetic algorithms.

Among them the most interesting ones are Normal Boundary Intersection NBI and Physical Programming PP. The reason why they are the most interesting in the context of this paper is that they can be used to calculate an even distribution of points as in [14] and [13]. The reason why most weighted methods do not work well, in terms of generating an even distribution of points on the pareto frontier, is examined in [3] where issues with concave solution sets are examined. Authors examine various cases of pareto frontier shape (including both concave and convex) and point out cases where most weighted, linear, methods cannot find points on concave parts of pareto frontiers.

Physical Programming [13] is the method which allows calculation of an even distribution of points across the pareto frontier. One of its benefits is that the decision maker does not have to specify any weights between functions. The decision maker expresses his preferences by giving bounds on resources within which he would like the resource consumption or the price to be. For example a decision maker can say that using 100 units of fuel is ideal, between 100 and 120 is desirable, between 120 and 160 is acceptable but undesirable and above 160 is unacceptable. PP uses this information, so the DM is required to provide these bounds on resources. There are eight classes of criteria classification in PP, divided into two main subclasses: soft and hard constraints. For soft constraints we can use the following: smaller is better, larger is better, value is better, range is better, which favours smaller, larger, exactly X, and any X within the range values respectively. similarly for the hard constraints we have must be smaller, must be larger, must be equal, must be in range. For each of the Soft classes the DM is required to specify six ranges of preferences: Ideal, Desirable, Tolerable, Undesirable, Highly Undesirable, Unacceptable. For the hard criteria only two ranges are defined: Acceptable and Unacceptable. Then, based on these preferences, PP uses the Linear Physical Programming Weight (LPPW) algorithm to compute weights. These weights are then used in a new LP problem which tries to minimize the deviation from the most desirable ranges. The actual algorithm for calculating these weights and then formulating the LP problem is different for each of the classes of criteria.

It is important to note that in [14] Messac and Mattson describe a slightly different way of using PP: Authors use equaly distributed weights, as opposed to using the method of calculating PP weights from the original paper, in order to generate an even distribution of pareto points. This approach is similar to what we are examining in this paper, as one of our approaches is to generate different weightings on objective functions and ask planners to generate solutions, hoping that they will be in different areas of the search space. A metric sensitive planner should be able to generate those solutions.

3.4. Presenting a Pareto Frontier

The presentation of a pareto frontier is also a challenge. Once all of the plans are generated and the trade-offs are known, the main concern is how to communicate the alternatives to the DM in a clear way, allowing them to see the trade-offs and make appropriate decisions. There has been a good progress in User Interface representation of solutions presented in [7] and [11], where authors are dealing with the multi-objective scheduling problem for observations using space telescopes. The difficulty of presenting the distribution of solutions starts as we introduce more and more dimensions. Dealing with visualization of up to three dimensional spaces is not very challenging, however, as the dimensionality increases it is harder to display the results. The approaches taken include projections of the pareto frontier onto lower dimensions and presenting them as plots or histograms of objective values, or as explicit values. All this combined in a clear GUI gives the DM a good understanding of the trade-offs involved and allows an informed decision.

4. Experiments

4.1. Experiment Set Up

The experiments will use a series of state of the art planners which have been selected based on their ability to work with numbers. We start by giving a brief description of the planners used in the experiments.

4.1.1. LPG-Metric

LPG [5] is a local search, stochastic planner. It creates its search space based on a graph with interleaved proposition and action layers called numerical action graph (NAG). Its heuristic consists of two elements, search cost and execution cost. Where search cost is a cost to resolve all inconsistencies created by inserting a new action estimated by solving a relaxed NAG. Execution cost is the total cost of executing actions in the plan and it represents plan quality. There are two weights on these two components which allow to trade-off finding solution quickly or searching for a good quality solution, depending on the need and constraints. LPG has been adopted in [15] to generate sets of plans. The change is to use ICP measure inside its heuristic instead of the standard execution cost.

4.1.2. MetricFF

MetricFF [8] is an extension of FF planner [9]. As an extension to a delete relaxation, removing all negative effects of actions, it treats all of the numerical effects as linear keeping its lower and upper bound. Therefore if at some point $x>2$ becomes true, it remains true until the end of the plan.

4.1.3. POPF

POPF [1] is a forward search planner which exploits some partial ordering of actions to avoid searching all orderings. Which means it does not enforce a strict total ordering on actions before the final stage of planning. For all facts and variables it keeps a list of propositions it has to support in order to execute the plan. While expanding a node, which is a partial-order plan it adds actions and creates new partial-order plans to which we could get from the current one.

4.1.4. LPRPG

LPRPG [2] uses relaxed planning graph (RPG) heuristics combined with linear programming (LP) methods. It solves a number of LP for every decision it makes to calculate bounds on resources and to improve its numeric reasoning. Thanks to solving the LP, LPRPG has more precise information about bounds on resources than other planners and therefore is designed for use in domains with numeric resource flaws.

4.2. Determining Metric Sensitivity of Planners

In these experiments we would like to evaluate a method of determining whether a planner is sensitive or insensitive to the change of metric. Our method is based on the definition of metric sensitivity as the ability to generate distinct solutions for a distinct objective functions. Following this definition we are going to run the planner asking it to generate a plan for a given problem in a domain multiple times. Each run we are going to change the objective function. After that, we are going to examine whether the plans generated follow the same pattern of change as our objective function.

The domain that we use is a modified version of DriverlogNumeric. In this domain we have the following objects: truck, driver and a package. We want to deliver packages to their final destination using trucks which have to be driven by drivers. Available actions are: load truck, unload truck, board truck, disembark truck, drive truck, walk. Where load and unload the truck require a package and a truck to be at the same location, similarly board and disembark truck requires a driver and truck to be at the same location, in addition, disembark requires the driver to be inside the truck. Driving requires a driver to be inside the truck, and the truck to be at the location where it is driving from. Walk requires the person to be at the location where it starts from. The effects of these actions are obvious.

Our extension of this domain includes changes like modifying single available truck into two categories: electric and diesel truck, adding a courier who can carry packages but only one at a time and walks much slower than trucks drive, representing a trade-off between objective function and plan length.

Example objective function for this file is:

```
(:metric minimize (+
  (* 4 (* (fuel-used) (fuel-used)))
  (* 6 (* (electricity-used) (electricity-used)))
)
```

For the experiment we have generated 11 different objective functions as follows:

$$\Theta = \alpha * (FuelUsed) + (1 - \alpha) * (ElectricityUsed)$$

The values for α are shown below the images. In Figure 2 for each objective function LPG was re-run multiple times and each of the dot on the diagram presents a plan and its position represents resources used by a plan after a single run. From Figure 2 it is clear that, for various α, LPG changed its behaviour depending on the weights of the objective function and, what is more, the change reflected the user's intentions expressed in the objective function. Therefore we can say that LPG is clearly a metric sensitive planner.

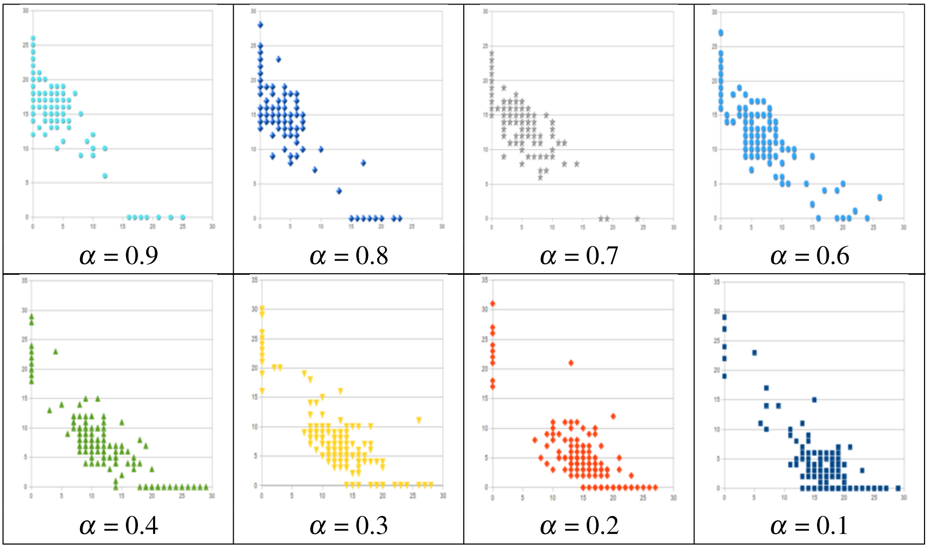

Figure 2. Representation of results for multiple runs of LPG on sets of objectives. α defines the weighting scheme (see text).

4.2.1. LPG Generating dDISTANTkSET

To compare the dDISTANTkSETs examined in [15] with the above results we have compared many sets and the non dominated set of plans from the previous experiments. The results can be seen in Figure 3.

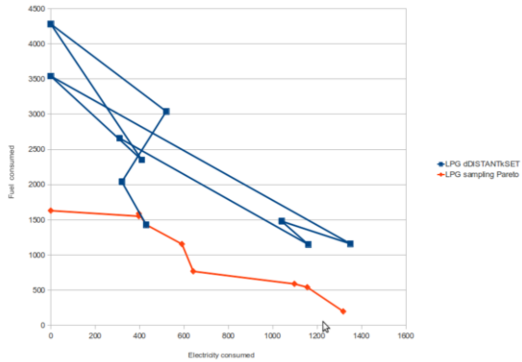

Figure 3. Pareto frontier generated by LPG using different weights, as in Figure 2, compared to a dDISTANTkSET which has plans connected in the order of them being found to illustrate how they are being 'pushed' away.

The dDISTANTkSET has been annotated with a line joining plans which have been found in order. This gives a feeling for how the search progressed towards finding the next solution. As we have noted before this method is trying to trade-off an aggregated cost of a plan and the execution time, but it does not focus on finding optimal plans, instead it finds a variety of plans. This drive to find different plans is reflected in lower quality of the plans as it is typically much easier to find different plans of a lower quality further from the pareto optimal frontier, where it is difficult to find plans.

4.2.2. MetricFF, POPF, LPRPG

In Figure 4 we can see that MetricFF generates the same plan no matter what the weights on objectives are. We also tried MetricFF with -O option, which emphasizes objectives, however, it also only generated one plan. MetricFF plan consumed 220 units of electricity and none of fuel and in the figure is represented by a mark on coordinates(220,0). The same happened for LPRPG, which generated the same plan in terms of resources used. When comparing the same plans in terms of actions used we could see differences and the standard approach would consider this two plans as different. However to our DM who is only concerned about resources used they are the same and therefore it is not desirable to ask him to choose between this two.

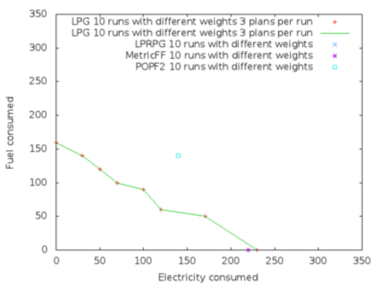

Figure 4. One point representing all plans generated by metricFF, LPRPG and POPF2 for all combinations of weights. In contrast to the whole pareto frontier generated by LPG.

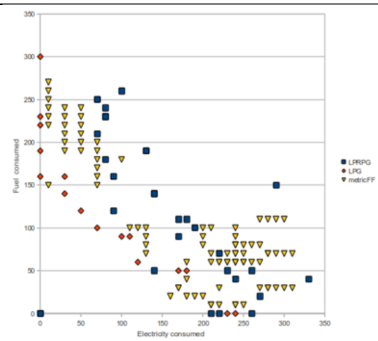

Figure 5. Sets of plans generated by previously metric insensitive planners like metricff and LPRPG using bounds on resources compared to metric sensitive LPG.

4.3. How Metric Insensitive Planners Can Behave in a Metric Sensitive Way

Based on the previous experiment we have identified that some planners are metric insensitive. Our work then focused on examining whether we can cause metricFF and LPRPG to behave in a way which simulates being metric sensitive. The approach taken is to impose bounds on resources it uses, including any of the lower or upper bounds on one or multiple objectives/resources at the same time. By limiting the amount of resource the planner could use, or by forcing it to use at least a certain amount of particular resource we aim to push it to explore different areas of the search space. Because these bounds are originating from metrics, if the planners behaviour changed it would mean its modified behaviour could be seen as metric sensitive. It is important to note that although we say that the planner becomes metric sensitive, it does not generate different solutions for different metric without translating these metrics into special bounds.

In this experiment we use lower bounds of minimum 0, 10, 20, 30, 40 and 50 units on fuel and electricity and all of their combinations which gave us 36 different bounds. Starting with (0,0) meaning use at least 0 fuel and 0 electricity units, then (0, 10), (0, 20) until (50, 50) meaning use at least 50 units of fuel and 50 units of electricity. We have experimented with numerous other ways of setting the lower and upper bounds, however, this method gave best results without a significant impact on the performance.

In Figure 5 we present a combined sets of results for planners which were unable to generate good quality sets in a weighted approach with the results for pareto frontier from LPG. MetricFF and LPRPG were given lower bounds on the resources which forced them to use minimum amounts of each resources and therefore do the trade-off. Combined results are presented in Figure 5.

It is clear that this approach of adding bounds on resources increased significantly the quality of the results achieved by these planners. They are also comparable with the plans found by LPG in the approach where multiple objectives were present.

4.4. How Metric Sensitivity Helps in Generating a Pareto Frontier

This experiment is meant to show whether metric sensitive planners can generate a well populated approximation of the pareto frontier only by changing the weights on objective functions. We have used the same domain and eleven problem files as before. By using different weights we expect to see points in different areas of the search space, which we think can form an approximation of the pareto frontier. For stochastic planners this process can be repeated and the best solutions for each weight is taken.

In order to generate pareto frontier we have generated sets of plans for each of the weights, merged the results together into a larger set visible on Figure 6 from which a subset of non dominated solutions (Definition 2) was selected.

The resulting set can be evaluated in many ways. For example as proposed in [15] by calculating and comparing the ICP values. At the moment we only focus on the method of generating pareto frontier and we do not try to evaluate it.

in Figure 7 we present pareto frontiers which are generated by a strongly metric sensitive planner, LPG, and by less sensitive, POPF, for comparison.

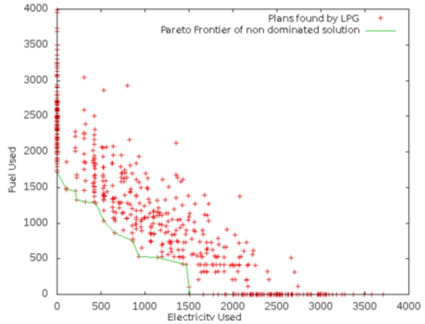

Figure 6. All plans generated by LPG using weights on objectives and the pareto frontier.

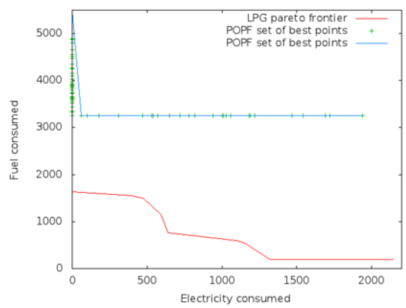

Figure 7. Comparison of results from the same experiment with weights between LPG and POPF.

5. Conclusion

As explained, metric sensitivity, as a key property of the planners, is a very important aspect of multi objective planning. Although, there has been some work done in related areas, there is still large scope for developing more advanced metrics to assess quality of results, quality of result sets and, most importantly, metric sensitivity of the planners. In this paper we wanted to show how important metric sensitivity is, how it can be used and how to evaluate whether a planner is or is not metric sensitive. It also turned out that planners, even metric sensitive ones, are still biased towards giving high priority to the plan length which is a built in metric in their heuristic.

In conclusion, we can say that LPG is the most metric sensitive planner among those which we have examined. From experiment 4.4 it is clear that LPG can be used to generate a good quality pareto frontier of plans, and what is more, this frontier is well populated with the plans being distributed across the whole length of it.

Many planners which we expected to exhibit more metric sensitivity (MetricFF, LPRPG and POPF), even though they turned out not to be metric sensitive, were able to generate a well distributed set of solutions when applied using lower bounds on resources. This is a novel idea and still needs more attention, however, it already produces good results by adding the notion of metric sensitivity to the planners.

References

[1] Amanda Coles, Andrew Coles, Maria Fox, and Derek Long, *Forward-Chaining Partial-Order Planning*, AAAI **20** (2010).

[2] Andrew Coles, Maria Fox, Derek Long, and Amanda Smith, *A Hybrid Relaxed Planning GraphâĂŞLP Heuristic for Numeric Planning Domains*, ICAPS (2008).

[3] Das and Dennis, *A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems.*, 1997.

[4] M. Fox and D. Long, *PDDL2.1 : An Extension to PDDL for Expressing Temporal Planning Domains*, Journal of Artificial Intelligence Research **20** (2003), 61–124.

[5] A Gerevini and I Serina, *LPG: A planner based on local search for planning graphs with action costs*, Proc. of AIPS-02 **1** (2002), 1.

[6] A.E. Gerevini, P. Haslum, D. Long, A. Saetti, and Y. Dimopoulos, *Deterministic Planning in the Fifth International Planning Competition: PDDL3 and Experimental Evaluation of the Planners*, Artifficial Intelligence **173** (2009), 1–63.

[7] Mark E. Giuliano, Reiko Rager, and Nazma Ferdous, *Towards a Heuristic for Scheduling the James Webb Space Telescope*, ICAPS (2007).

[8] J. Hoffmann, *The Metric-FF Planning System: Translating Ignoring Delete Lists to Numeric State Variables*, Journal of Artifcial Intelligence Research **20** (2003), 291–341.

[9] J. Hoffmann and B. Nebel, *The FF Planning System: Fast Plan Generation Through Heuristic Search*, Journal of Artificial Intelligence Research **14** (2001), 253–302.

[10] C.L. Hwang, S. R.Paidy, and K. Yoon, *Mathematical programing with multiple objectives: a tutorial*.

[11] Mark D. Johnston and Mark Giuliano, *Multi-Objective Scheduling for the Cluster II Constellation*, (2011).

[12] R.T. Marler and J.S.Arora., *Survey of multi-objective optimization methods for engineering*, 2004.

[13] Achille Messac, Surendra M. Gupta, and Burak Akbulut, *Linear Physical Programming: A New Approach to Multiple Objective Optimization*, 1996.

[14] A. Messsac and C.A.Mattson, *Generating Well-Distributed Sets of Pareto Points for engineering Design Using Physical Programming.*, 2001.

[15] T. Nguyen, M. Do, A. Gerevini, I. Serina, B. Srivastava, and S.Kambahampati, *Planning with Partial preference Models*, TechnicalReport (2011).

[16] Tuan A. Nguyen, Minh B. Do, Subbarao Kambhampati, and Biplav Srivastava, *Planning with Partial preference Models*, International Joint Conferences on Artificial Intelligence IJCAI **1** (2009), 1.

[17] S.Zionts and J. Wallenius, *An Interactive Programming Method for solving the Multiple Criteria Problem*, 1975.

[18] S.V. Utyuzhnikov, P. Fantini, and M.D. Guenov., *A method for generating a well distributed Pareto set in nonlinear multiobjective optimization*, 2007.