Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Are Automated Planners up to Solve Real Problems?

Fernando Moreira Sette*, Tiago Stegun Vaquero*, Song Won Park, Jose Reinaldo Silva***

**DesignLab, Mechatronic and Mechanical Systems Department*
Escola Politécnica – University of Sao Paulo, Brazil.
(e-mail: {fernando.sette, tiago.vaquero}@poli.usp.br, reinaldo@usp.br)
***Chemical Engineering Department. University of Sao Paulo. Brazil.*
(e-mail: sonwpark@usp.br)

Abstract: It is a well known fact that the AI planning community is very committed to apply the developments already achieved in this area to real complex applications. However realistic planning problems bring great challenges not only for the designers during design processes but also for the automated planners during the planning process itself. In addition, it is quite common to face issues about whether the available planners will be up to solve the problem being modeled during the initial design stages. In this paper we present the experience, results and issues that emerged from testing the performance of the recent planners when solving a real and complex problem such as the planning of daily activities of a petroleum plant for docking, storing and distributing oil. Due to the complexity of this real planning problem, the KE tool itSIMPLE was used in order to support all the design processes such as specification, modeling and domain model analysis that resulted in a PDDL model, automatically generated by the tool, which was used as input for planners. In addition, we present the main modeling process performed for the domain model construction.

1. INTRODUCTION

The recent efficiency improvement, the rising demand for planning systems and the development of Knowledge Engineering (KE) tools (making it practical to model complex and real domains) have become a great motivation to try to apply all achievements already conquered by the Artificial Intelligence Planning community in real life applications. However, in addition to the challenges involved in modeling realistic domains, there is a great uncertainty, throughout the design process, about the ability of current available automated planners to solve the problem being modeled. This can be prejudicial when the designer feels forced to reduce the model to a point where the solution obtained is no longer reliable in a realistic context. The main purpose of this work is to share the experience of designing and testing a real planning domain while presenting an evaluation of the feasibility of using modern AI planning techniques and tools in solving real problems.

The real problem presented in this paper deals with the planning of the daily activities of a petroleum plant for docking, storing and distributing oil. The planning of these operations is very important to the functioning of refineries and constitutes a complex problem of difficult mathematical modeling (Dahal *et al.,* 2003). When planning over this problem engineers must deal with tankers allocation, docking scheduling, tank volume control, crude oil storage with price maximization (avoiding mixing certain types of crude oils) and minimization of costs. In fact, this problem presents many challenges, such as resource allocation, sequencing, scheduling and optimization, among others.

Therefore, given the relevance of fluid transportation problems in modern industry (approximately 25% of all load transported in the US occurs through pipelines (Más and Pinto, 2003) and the complexities involved in this problem, the evaluation of the performance of current available automated planning techniques and tools when dealing with such a domain is a very good way to measure the applicability of planning to real world problems.

Since not all planners available were able to deal with all the requirements involved in this domain, only three planners were evaluated in the testing stage of the design process: SGPlan (Hsu *et al.,* 2006), Metric-FF (Hoffman, 2003) and MIPS-XXL (Edelkamp *et al.*, 2006), that were able to deal with the numeric statements and the minimization aspect of the problem.

The requirements phases and domain modeling process was performed using the KE tool itSIMPLE - Integrated Tools Software Interface for Modeling PLanning Environment (Vaquero *et al.*, 2007) (Vaquero *et al.*, 2006) - in which all the model was built utilizing the UML – Unified Modeling Language (OMG, 2001) - in a Planning approach. Due to the size and complexity of this real problem it is indeed necessary to use tools that provide means of dealing with the main design process such as requirements acquisition, modeling and analysis. Actually, designing the domain in PDDL - Planning Domain Definition Language (Fox and Long, 2003) - from scratch would have proven extremely difficult and time consuming. Also, we believe that using such a KE tool it contributes to finding better modeling solutions as well as to identifying relevant domain issues and features that otherwise could not be recognized by a totally action driven specification.

This paper is structured as follows. First, we present the domain, its restrictions and requirements. Then we present the domain model designed using the KE tool itSIMPLE. Next, we depict the studied scenarios that were tested with the planners. This paper ends with the description of the experimental results, some discussions and the conclusion.

2. OIL SUPPLY AS A PLANNING/SCHEDULING PROBLEM

Operations with crude oil involve the unloading of tankers in docking stations into distribution tanks, and the supply of refineries. Since the refineries are constantly consuming oil, the plan must guarantee that, at all moments, the amount of oil in the refineries remains above a minimum level, while minimizing the cost of distribution.

Nowadays, most research work done in this area has utilized mathematical programming where the models are adapted to mixed-integer linear programming (MILP) or mixed-integer non-linear programming (MINLP) to find solutions to this problem. However, current methods have failed to show feasible solutions or require a great amount of time to solve these problems. Furthermore, MILP methods require the use of linearization, which leads to flaws in the final solutions, while the discretization necessary in MINLP methods greatly increases the size of the problem (Li *et al*., 2005). Therefore, there is no reliable efficient and robust algorithm for this real, and very important, problem in current literature (Li *et al*., 2005).

In this work, a real problem encountered in one of the main oil supply distribution complexes of Brazil will be investigated under the automated planning perspective. The domain description and requirements was based on the work of Mas and Pinto (2003) and the information provided by Petroleo Brasileiro S.A. (Petrobras), the main petroleum producer and distributor in Brazil.

Crude oil is processed in four refineries in the State of Sao Paulo (Brazil): Paulinia (REPLAN), Sao Jose dos Campos (REVAP), Cubatao (RPBC) and Capuava (RECAP), which are supplied through a pipeline network that leaves the Sao Sebastiao terminal (GEBAST). The system also contains two intermediate substations (SEBAT, in Cubatao, and SEGUA, in Guararema), as well as pumping stations in Rio Prado and Guaratuba. All the crude oil that is consumed by the State of Sao Paulo comes through GEBAST and is distributed by two pipelines: OSVAT and OSBAT. This system is detailed in Fig. 1.

This work considers only the planning of the Sao Sebastiao terminal operations: docking of oil tankers and storage and distribution of crude oil to refineries.

2.1 Domain and Problem Requirements

The distribution complex considered consists of a port, refineries and pipelines that carry the oil to the refineries where it will be processed. The port contains piers, tanks, and an internal pipeline that connects the two structures. This last item has already been subject of study in the planning community, having appeared in the International Conference on Automated Planning and Scheduling ICAPS'04 as a domain in the fourth International Planning Competition IPC'04 (Hoffmann *et al.,* 2004). However, while this problem is very operational in nature, this paper is concerned with a more strategic issue that is the planning and scheduling of crude oil distribution in order to maximize profit leaving the internal pipeline issue apart.

The planning and scheduling of port activities involves several activities such as: assignment of tanks to piers, unloading of the tankers to the tanks in the terminal and unloading of the terminal tanks to the pipelines (Mas and Pinto, 2003).

Tanker requirements: The crude oil arrives at GEBAST through oil tankers, which are unloaded at the docking stations and stored in the tanks of GEBAST. Each docking station has a limitation regarding the size of the tankers it can receive.

Furthermore, this unloading operation has to be done quickly and efficiently, since there are severe overstay costs in this operation. Each tanker has a limited time that it can stay docked in the pier and unload without paying overstay costs. Therefore, the planning of this operation should respect this period whenever possible.

Finally, every tanker takes a certain time to dock and to leave the port. In practice, this means that, after the order to dock is given, a period must be waited before unloading operations begin and that a docking station will only be able to receive another vessel a certain period after the exit order is issued to the tanker currently occupying it.

Tank requirements: Petrobras processes several different types of oil in its refineries. Since reserving a tank for each oil type is not practical, these are grouped into classes. The crude oil types that belong to a class can be mixed together without losing value (Más and Pinto, 2003).

At a given moment, a tank can be in either one of three states: inoperative, loading or unloading. Under no circumstance can a tank be unloading and loading simultaneously (Mas and Pinto, 2003). Furthermore, there are some restrictions concerning the presence of brine in the tankers inventories. Since every oil type unloaded at Sao Sebastiao contains brine (even after separation in petroleum production platforms), the tanks must undergo a settling period (during which the tank remains inoperative), having received crude oil from a tanker, before it can send oil to the refineries. During this period, the brine settles in the bottom of the tank. This is done in the tanks of GEBAST because it is not desirable to transport brine through the pipelines or send it the refineries (Más and Pinto, 2003).

In order to prevent the accumulation of volatile components, the tanks operate using a floating roof system. Since a minimum safety level is required, in order o avoid damage to these structures, the tanks cannot ever be fully unloaded

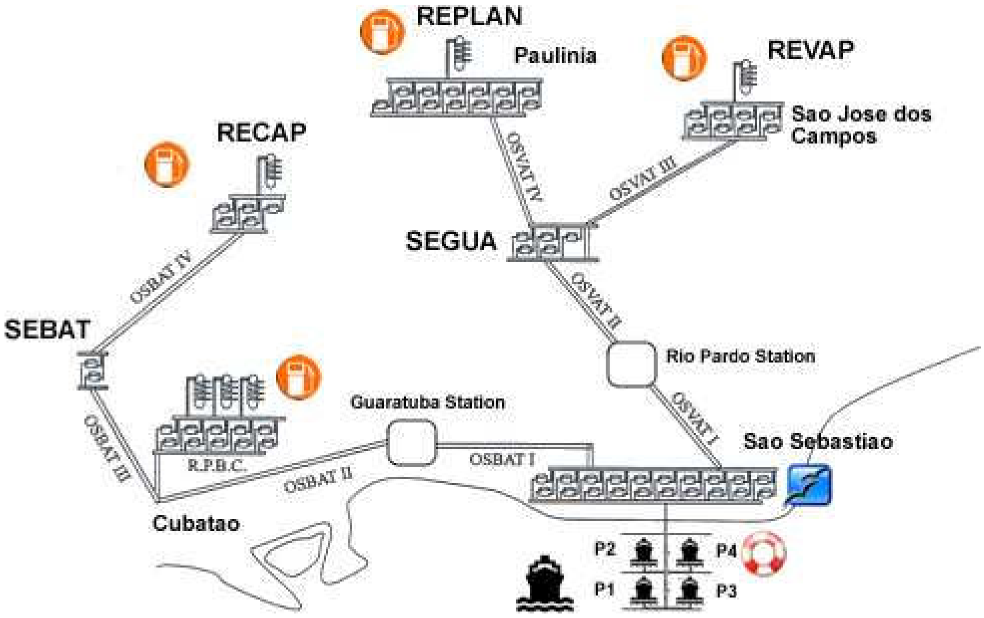

Fig. 1. Crude oil distribution infrastructure of Petrobras in the State of Sao Paulo, Brazil.

(Mas and Pinto, 2003). This restriction is, usually, about two meters, which represent about 15% of the total capacity. Therefore, each tank has a maximum and minimum capacity that must be respected in the planning of their operation.

Pipeline requirements: The pipelines are used to send oil from the terminal to the refineries that will process it and are able to transport, simultaneously, more than one crude oil type. During this operation, an interface forms between two different oil types that results in a loss of their properties (Más and Pinto, 2003). Furthermore, a pipeline cannot unload tanks simultaneously.

Refinery requirements: The refineries have maximum and minimum capacity restrictions that must be respected throughout their operation. However, a model of the refinery operation will not be considered and they will be modeled as continuous consumers of crude oil. It will be assumed that, in the short term, the refineries will have established an average rate of consumption of crude oil.

2.2 The Domain Modeling Process with itSIMPLE

itSIMPLE (Vaquero *et al.*, 2007) (Vaquero *et al.*, 2006) proposes a special planning approach based on UML (OMG, 2001), named UML.P, to be used during planning domain specification and modeling process. This method utilizes UML diagrams, such as Use Case, Class, StateChart and Object Diagrams, to aid the designer in the requirements elicitation and domain modeling stages. The following descriptions of these diagrams illustrate the construction of the model using this tool.

In this work, the model does not consider time constraints involved in the operations of this system. This restriction does not allow the modeling of the refineries operations (since they behave as continuous oil consumers) and costs that involve time (such as overstay costs). However, even though time related costs have an important impact on the overall economic result of the operation, most of revenue in this operation is related to the correct allocation of oil into storage tanks. Since different classes have different values, choosing the correct class in which to store the oils to maximize their values and minimize other non time-related costs (such as interface costs) is an essential part of the planning of oil supply. Models considering time will be left for future works.

Finally, since most of available general planners are only able to deal with PDDL, the generated UML model was translated to this language in order to test the performance of such available planning techniques. In fact, itSIMPLE is capable of automatically translating the UML.P model into PDDL and for the current planning problem it was generated a PDDL model with features of the PDDL2.1 level 2 (Fox and Long, 2003). The resulting PDDL model will be made freely available in (DesignLab, 2008).

2.3 Domain Modeling

Use Case diagram: The Use Case diagram models the domain in the highest abstraction level where the domain scope is firstly defined. This diagram facilitates the unification of the different viewpoints involved. The Use Case diagram for the Sao Sebastiao terminal oil distribution activities is showed in Fig. 2. The domain requirements were represented in the elements in the diagram where each use case receives a full description, pre and post condition, constraints, invariants, flow events and other relevant information.

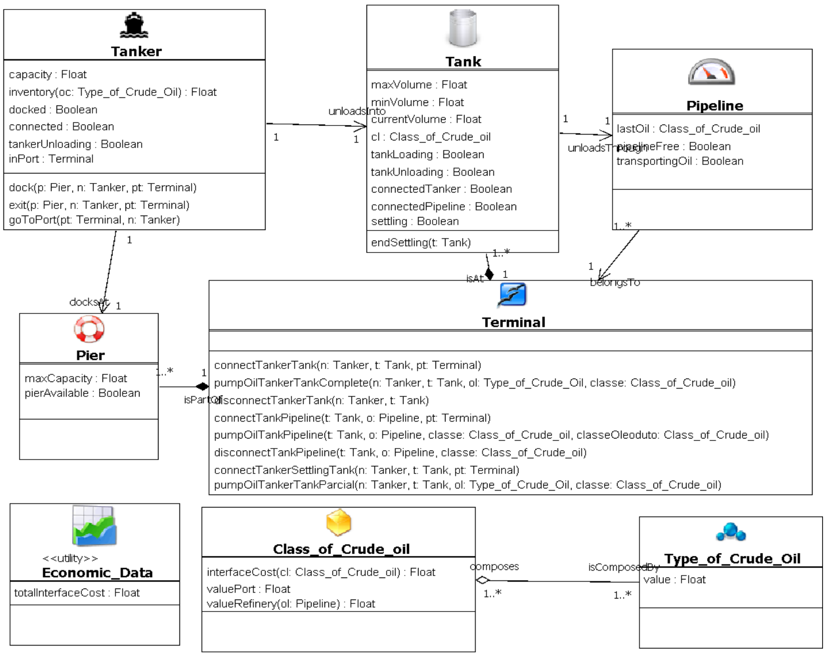

Fig. 3. Class diagram of the Oil Supply domain

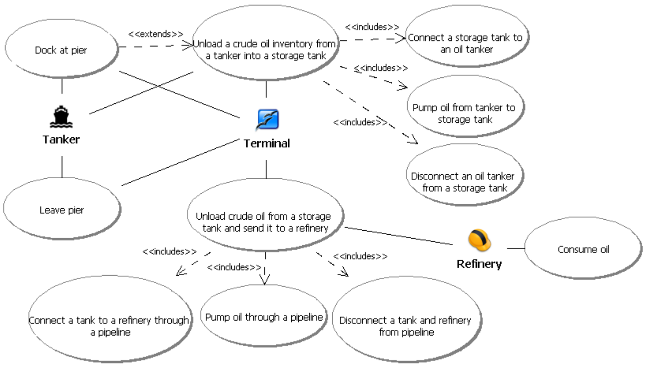

Fig. 2. Use Case diagram of the Oil Supply domain

As seen in Fig. 2, this oil distribution system, which is centered at the terminal, possesses three independent agents (actors in UML.P): tanker, the terminal itself and the refineries. The actors interact together to perform the tasks required to take the oil from the tankers and deliver it to the refineries.

Class diagram: The Class diagram is a representation of the static structure of the planning domain. This diagram shows the existing entities, their relationships, their features, operators (actions) and constraints. Class attributes and associations between classes give a visual notion of the model semantic. It is important to mention that the class diagram does not represent a particular planning problem, but a set o planning problems.

Fig. 3 shows the class diagram designed for the oil distribution problem ate Sao Sebastiao terminal. The diagram consists of eight classes that model all the entities relevant to the real problem being modeled. The Economic_Results class is a utility class that stores variables that are relevant to all other classes in the model such as interface costs. In this particular case, it corresponds to cost and revenue variables which are used as metric for the optimization of profit.

As mentioned earlier, the refineries are not modeled in the class diagram, since time has not yet been included in the modeling of the problem.

StateChart diagram: The dynamic behavior of actions is specified in the StateChart diagram (also called State Machine diagram), where it is possible to define the pre and pos conditions for the operators defined in the class diagram. This diagram is very useful when representing system entities that perform dynamic behavior. In the itSIMPLE tool all the pre and post condition are defined by using the formal constraint language called OCL - Object Constraint Language (OMG, 2003) - , which is a predefined language of the UML.

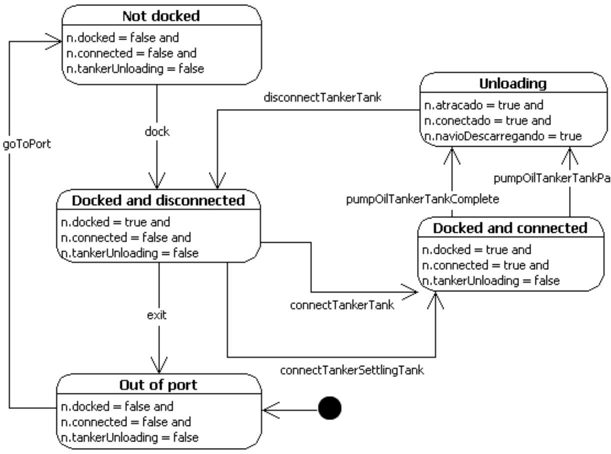

Fig. 4. StateChart diagram the class Tanker

Usually every class in Class diagram (especially those that change states during a plan execution) has its own StateChart diagram. One StateChart diagram does not intend to specify all changes caused by an action. Instead, it details only the changes that it causes in an object of a specific class. Fig. 4 shows the StateChart diagram for the class Tanker.

2.4 Problem Modeling

In itSIMPLE problems are modeled using the object diagrams. All the planning problems for the current Oil Supply domain were modeled in these diagrams.

Object diagram: A problem statement in a planning domain is characterized by a situation where only two points are known: the initial and goal state. The diagram itSIMPLE uses to describe these two states is called Object Diagram or Snapshots. A snapshot is nothing more than a picture of the system at a specific state. It is also seen as an instantiation of the domain structure defined in previous diagrams. The instantiation defines how many objects are in the problem; which are their classes; what are the values of each object attribute and how they are related with each other.

A planning problem is composed by two Object Diagrams: one describing an initial state and, another, the goal state (partial or complete). Fig. 5, shows the initial snapshot for a planning problem example used in this work during the test with planners stage, while Fig. 6 shows the goal state for the same problem. Indeed, it would be an arduous task to provide a complete definition of the goal state for the problem studied in this paper and, therefore, partial goal stated were used. The partial goal states consisted of all tankers unloaded (zero inventory) and undocked. The states of all other objects in the problem were left undefined.

The following section describes the experiments done with the modeled planning problems.

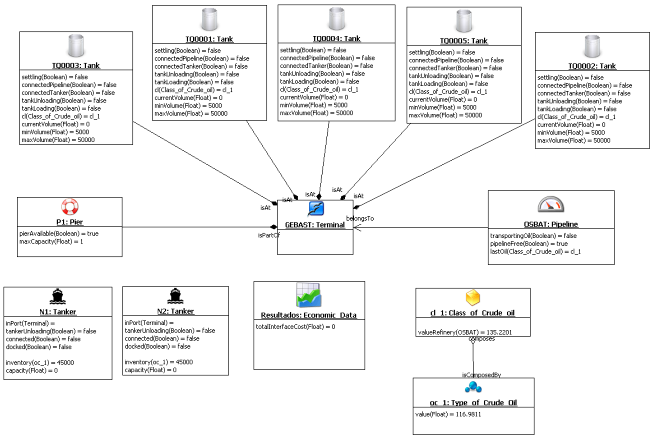

Fig. 5. The initial state of a problem as a snapshot

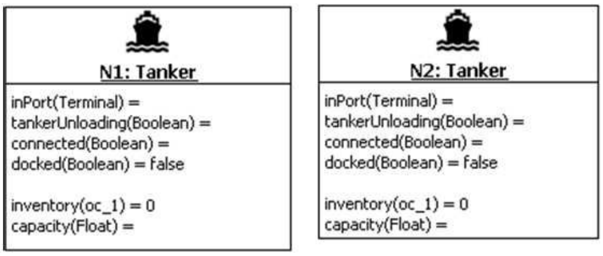

Fig. 6. The goal state of a problem as a snapshot

3. EXPERIMENTS

The domain presented in this paper has requirements that go beyond the classical STRIPS - Stanford Research Institute Problem Solver - problems. Therefore, the experiments could not be performed with all available planners since many of them do not deal with numerical constraints. The planners considered in this work were those that participated in the competitions IPC'02, IPC'04 and IPC'06 in the temporal/metric category. Three high performance planners were chosen and tested: Metric-FF (Hoffman, 2003), SGPlan (Hsu *et al.*, 2006) and MIPS-XXL (Edelkamp *et al.*, 2006).

3.1 Scenarios

Two scenarios were considered in this work: one relates to a real problem example and the other consists of made up simpler problems. The latter is used to find the planner response to certain variables (such as the size of the crude oil inventories in oil tankers), while the former measures how the planner behave as we come closer to a real problem.

Scenario one: The real problem presented in this paper is an example of a situation in the Sao Sebastiao terminal considering a timeframe of one week: thirteen tankers, carrying fourteen different oil types, are scheduled to arrive in the port to be unloaded.

The port infrastructure possesses eighteen tanks available to store the crude oil inventory arriving. These crude oils are grouped in seven different classes, each tank being able to store only one class of crude oil (one oil type can be a member of more than one class).

It is worth mentioning that, even though the order in which the tankers arrive at the port is predefined, this constraint is not added in the model in order to find the sequence of tanker arrival that will maximize profit.

The unloading of tanker is performed in the four docking stations in the terminal. This oil is then sent to refineries through two pipelines that leave the terminal.

The purpose of this scenario is to evaluate how the available planners respond as we approach a real situation. To this end, thirteen problems were created where the number of oil tankers varies from one to thirteen. Also, an optimization objective is included concerning the minimization of the interface costs during the distribution operation.

Table 1. Realistic problem without metric

# of Tanks	# of Tankers	Metric FF		SGPLAN		MIPS	
		Actions	Time	Actions	Time	Action	Time
18	1	7	0,46	7	0,46	13	1,14
18	2	17	1,82	17	0,97	25	49,84
18	3	30	3,24	31	4,66	TIME	TIME
18	4	X	X	62	54,18	TIME	TIME
18	5	FLUENTS	FLUENTS	75	120,48	TIME	TIME
18	6	FLUENTS	FLUENTS	78	42,74	TIME	TIME
18	7	FLUENTS	FLUENTS	TIME	TIME	TIME	TIME
18	8	FLUENTS	FLUENTS	97	74,43	TIME	TIME
18	9	FLUENTS	FLUENTS	TIME	TIME	TIME	TIME
18	10	FLUENTS	FLUENTS	TIME	TIME	TIME	TIME
18	11	FLUENTS	FLUENTS	115	303,78	TIME	TIME
18	12	FLUENTS	FLUENTS	126	386,72	TIME	TIME
18	13	FLUENTS	FLUENTS	132	478,73	TIME	TIME

Table 2. Realistic problem with metric

# of Tankers	# of Tanks	Metric FF			SGPLAN			MIPS		
		Actions	Time	Metric	Actions	Time	Metric	Actions	Time	Metric
1	18	-	-	-	7	0,34	0,00	X	X	X
2	18	-	-	-	17	1,86	0,00	X	X	X
3	18	-	-	-	31	4,70	2,20	X	X	X
4	18	-	-	-	62	55,13	5,66	X	X	X
5	18	-	-	-	75	121,00	3,15	X	X	X
6	18	-	-	-	78	42,88	4,17	X	X	X
7	18	-	-	-	TIME	TIME	TIME	X	X	X
8	18	-	-	-	97	74,43	8,805	X	X	X
9	18	-	-	-	TIME	TIME	TIME	X	X	X
10	18	-	-	-	TIME	TIME	TIME	X	X	X
11	18	-	-	-	115	303,78	-	X	X	X
12	18	-	-	-	126	386,72	-	X	X	X
13	18	-	-	-	132	478,73	-	X	X	X

Table 3. Simple problem with varying volume of tanker inventory

# Tanks	# Tankers	Oil inventory in tanker (m^3)	Metric FF		SGPLAN		MIPS	
			Actions	Time	Actions	Time	Actions	Time
5	6	45000	41	0,48	38	0,34	38	0,71
5	6	90000	83	12,59	87	0,23	83	88,63
5	6	135000	TIME	TIME	131	0,85	TIME	TIME
5	6	180000	TIME	TIME	197	2,79	TIME	TIME
5	6	225000	TIME	TIME	248	9,75	TIME	TIME
5	6	270000	TIME	TIME	314	11,73	TIME	TIME
5	6	315000	TIME	TIME	332	7,10	TIME	TIME
5	6	360000	TIME	TIME	395	20,16	TIME	TIME
5	6	405000	TIME	TIME	413	14,53	TIME	TIME
5	6	450000	TIME	TIME	416	15,35	TIME	TIME

Scenario two: These are simpler made up scenarios designed to measure how the planner behaves as the volume of oil in the tankers varies. As the crude oil inventory in these vessels increase, the planner is forced to perform more actions in order to unload all the tankers in the port. This happens because, since tanks have a limited capacity, it becomes necessary to unload the tanks more often, increasing the number of actions necessary to find a solution.

The problems in this scenario consist of three tankers carrying equal volumes of a crude oil type. The terminal possesses one docking station and five tanks to perform the unloading operations. All tanks have the same capacity and store the same class of crude oil.

The variable which varies among the problems in this scenario is the number of tanks necessary to unload all the inventory of one tanker. Since the inventories of all vessels and capacity of all tankers, this measures the minimum number of tank unloading operations that must be performed in order to unload all tanker fully. This variable is varied from one to ten tanks in the problems present in this scenario.

3.2 Experiment Results

The planners were run on Intel Pentium IV computer with 1.0 gigabyte of memory, running Fedora Linux. Tables 1 and 2 show the results for Scenario 1, and Table 3 for Scenario 2. All tables show the number of tanks and oil tankers in the problem, the number of actions in the plan and the computational time used by the planner to find a solution. All experiments were terminated once ten minutes had elapsed and no solution been found: a “TIME” symbol in the tables indicates when this happened. The cases where the planner terminated without finding a solution to the problem are designated by an “X” symbol.

In addition, Table 2 shows results from problems that include the metric of the total interface cost minimization and Table 3 the shows the performance results highlighting the volume of the tankers inventory in each problem tanks.

Table 1 shows that Metric-FF and MIPS-XXL demonstrated difficulties solving the majority of the problems for realistic problems without metric, while SGPlan managed to come up with a solution for practically every problem proposed in this experiment. However, finding just any plan is not useful for this particular application, since the problem is to find a solution that will maximize profits.

When metric is added to the problem, even though SGPlan continues to find solutions it is clear that it is not performing any optimization while solving the problem. The computing time and number of actions in the solutions remains the same. Metric-FF gave error messages even before beginning to compute a solution and MILP-XXL was unable to solve any of the problems when metric was involved. On this last case, the planner had an internal limitation of 300 to find the solution. However, since the simpler problem in this scenario was solved in under 0.5 seconds by both Metric-FF and SGPlan, 300 seconds is a remarkable increase in time, which shows how the optimization requirement greatly affects the performance of planners.

The metric utilized in this work was the total interface cost (which incurs when different oil classes are transported in the same pipeline) of the operation. In some cases, all the tankers could be emptied without the need to unload any tanks. In these situations, the metric returns zero since no oil went through the pipelines, but still solving the problem specified.

These results also show that SGPlan had difficulties solving particular problems. This can be explained by the fact that this planner considers that many planning applications have a cluster structure of the constraints and, then, the goal can be partitioned into sub-goals based on these clusters. Each sub-goal is solved by a modified version of the FF planner and then a global analysis resolves global constraint inconsistencies by a penalty formulation.

The problems that caused TIME in SGPLAN execution could have many causes. One of them is that these problems can affect the clustering of constraint and consequently the sub-problems inherit some partition mistakes that disable the system to find a solution in the specified time. Another possible explanation for the SGPLAN breaks can be the use of Metric-FF to solve sub-problems. The Metric-FF does not solve the same problem in our experimentation and it has a considerable probability to be unable to solve a sub problem of such problem. Even more so if we consider that the partition could have a mistake.

Metric-FF follows the principles of the FF planner. FF planner constructs an entire net with all predicates (fluents) and all actions as well as the links among them. This entire net, can result in a memory overflow for problems with many objects. That is a possible cause of the FLUENTS in the experimentation.

4. DISCUSSION AND CONCLUSION

In this work, a real complex planning problem, such as the planning/scheduling of the daily activities of a crude oil distribution plant, was presented and modeled in order to investigate the ability of current available planners to solve such problems. Since this is a large and very complex problem, the requirement, specification and modeling processes were done with the support of the KE tool, itSIMPLE (Vaquero *et al.,* 2007).

Due to the domain model properties few planners were able to deal with all the characteristics involved. For this reason only three planners were investigated: Metric-FF, SGPlan and MILP-XXL. Generally, these planners did not show a successful performance when solving realistic problems in the studied domain. They did not have satisfactory results on either time of response or plan quality or both. The results were good only when the problems were relatively simple, small and non-realistic problems. It seems that domain models that use numeric variables to a great extent cause a great impact in the planning process when compared with STRIPS-like domains. In fact, it is common to find real planning problems that possess important and essential numeric variables, especially those which requires optimization of certain parameters.

The results obtained from the scenarios indicate that, in order to get good results from available planners, it is necessary to reduce the domain in such a way that the gap between the real requirements and the model is greatly extended. In addition, the presented domain is already a simplification of the real problem done for this work, since time, which makes the model much more complex, was omitted.

Finally, general planners can be an important tool for domain model testing in order to evaluate certain properties of the domain and also to perform preliminary evaluation concerning the problem solutions. However, in order to go a step further in the design process, the need for a dedicated planner with specific heuristics, extracted from the model, arises.

REFERENCES

Dahal, K., S. Galloway, G. Burt, J. McDonald and I. Hopkins (2003). Port system simulation facility with an optimization capability. In: *International Journal of Computational Intelligence and Applications*, vol. 3, no. 4, pp. 395-410.

DesignLab (2008). http://designlab.grupoe.com.br. Access: February 10th 2008.

Edelkamp, S., S. Jabbar, and M. Naizih (2006). Large-scale optimal PDDL3 planning with MIPS-XXL. In: *5th International Planning Competition Booklet (IPC-2006)*, pp. 28– 30, Lake District, England, July 2006.

Fox, M., and D. Long (2003). Pddl2.1: An extension of pddl for expressing temporal planning domains. In: *Journal of Artificial Intelligence Research (JAIR),* 20:61–124.

Hoffmann, J. (2003). The Metric-FF planning system: Translating "ignoring delete lists" to numeric state variables. In: *Journal of Artificial Intelligence Research*. Accepted for special issue on the 3rd International Planning Competition.

Hoffmann, J., S. Edelkamp, R. Englert, F. Liporace, S. Thiébaux and S. Trüg. (2004). Towards Realistic Benchmarks for Planning: the Domains Used in the Classical Part of IPC-4. In: *Booklet of the Fourth International Planning Competition*, pages 7-14, June 2004.

Hsu, C.W., B. W. Wah, R. Huang, and Y. X. Chen (2006). New Features in SGPlan for Handling Soft Constraints and Goal Preferences in PDDL3.0. In: *Proc. Fifth International Planning Competition, International Conf. on Automated Planning and Scheduling ICAPS'06*, June 2006.

Li, J., L. Wenkai, I. A. Karimi and R. Srinivasan (2005). Robust and Efficient Algorithm for Optimizing Crude Oil Operations, In: *American Institute of Chemical Engineers Annual Meeting*.

Más, R., and Pinto (2003). A Mixed-Integer Optimization Strategy for Oil Supply in Distribution Complexes. In: *Optimization and Engineering*, vol. 4, no. 1-2, pp. 23-64, Jun. 2003.

OMG (2001). OMG Unified Modeling Language Specification, Version 1.4.

OMG (2003). OMG Unified Modeling Language Specification - Object Constraint Language, Version 2.0.

Vaquero, T. S. , V. M. C. Romero, F. Tonidandel, and J. R. Silva (2007). itSIMPLE 2.0: An Integrated Tool for Designing Planning Domains. *In: The International Conference on Automated Planning & Scheduling ICAPS '07*, Providence, Rhode Island, USA.

Vaquero, T. S., F. Tonidandel and J. R. Silva (2006). On the Use of UML.P for Modeling a Real Application as a Planning Problem. In: The International Conference on Automated Planning and Scheduling ICAPS'06, Cumbria, UK.