Dynamic Schedule Management: Lessons from the Air Campaign Planning Domain *

Brian Drabble and Najam-ul Haq

Computational Intelligence Research Laboratory,
1269, University of Oregon,
Eugene, OR 97403
drabble,haqn@cirl.uoregon.edu

Abstract. This paper describes the Dynamic Execution Order Scheduling (DEOS) system that has been developed to handle highly dynamic and interactive scheduling domains. Unlike typical scheduling problems which have a static task list, DEOS is able to handle dynamic task lists in which tasks are added, deleted and modified "on the fly" DEOS is also able to handle tasks with uncertain and/or probabilistic outcomes. DEOS extends the current scheduling paradigm to allow tasking in dynamic and uncertain environments by viewing the planning and scheduling tasks as being integrated and evolving entities. DEOS has been successfully applied to the domains of Air Campaign Planning (ACP) and Intelligence, Surveillance and Reconnaissance (ISR) management. The paper provides an overview of the dynamic task model and the "penalty box" scheduling algorithm which was developed to provide robust solutions to over constrained scheduling problems. The basic algorithm is described together with extensions to handle flexible time constraints.

1 Introduction

This paper describes the Dynamic Execution Order Scheduling DEOS system that has been developed to handle highly dynamic and interactive scheduling problems. Unlike typical scheduling problems which have a static task list, DEOS is able to handle dynamic task lists in which tasks are added, deleted and modified "on the fly". In addition, the dynamic tasking model used by the DEOS system allows it to handle tasks with uncertain and probabilistic outcomes. This allows DEOS to tackle a wider range of problems than possible with previous approaches.

* This research is supported by DARPA Contract: DABT63-98-C-0069 "Intelligent Workflow for Collection Management" and Contract: F30602-97-1-0294 "Understanding and Exploiting Hierarchy". The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either express or implied, of DARPA, Rome Laboratory or the U.S. Government.

Several systems [11, 6, 10] have attempted to solve problems in domains in which there are defined time bounds on activities or where an activity's outcome follows some predictable distribution. For example, in semi-conductor manufacturing a machine may have a failure rate of between 0.1% and 0.5% depending on the chip being manufactured. It may also be the case that some steps need to be re-executed to deal with failures and reworking, e.g. most of the failed chips can be fixed if they pass through steps 112 through 118 again. Systems such as CASPER [3] and CPEF [9] have also attempted to address the planning and execution problem. However, neither system has taken a resource centered optimization approach and neither has attempted to coordinate planning functions across distributed platforms. While these techniques have been successful in domains with limited amounts of uncertainty they are totally unsuitable for dealing with domains such as ACP that contain large amounts of uncertainty (e.g., partially order activities, activities with unknown durations, unexpected outcomes, new requirements) and probabilities (e.g., expected aircraft attrition rates, target damage, locations of enemy forces). The problem is further complicated by the distributed nature of the planning process in which different aspects of the plan are generated and maintained by separate planning cells (e.g. logistics, airborne tankers,, maintenance). The problem becomes one of optimally putting together many different scheduling pieces and monitoring their dependencies and requirements over time.

One of the key aspects of the ACP process is the scheduling of aircraft and weapons to targets (i.e., how many aircraft, of what type, carrying which weapons are assigned to the target). This is a very complex problem as it contains large numbers of different types of constraints (e.g., time, user priority, weight of effort [1], phasing [2], resources). The assignment problems needs to address three major concerns:

1. Identifying trade-offs between different aircraft assignments. For example, a mission's success can be increased if it has fighter escort but these same fighter aircraft could be employed on other bombing missions. If the optimization criteria is to minimize the schedule's makespan and to maximize mission success then the choice of whether or not fighter aircraft are assigned becomes an important trade-off.
2. Identifying the optimal set of targets which can be attacked with the resources available. This requires the scheduler to identify a subset of the targets that can be successfully assigned and to ensure the reasons why targets that are unassigned are fed back to the human planners. This allows for the development of more robust schedules (i.e., ones with a higher probability of succeeding) than previously available to USAF planners. In many cases the human planners would sooner have a schedule that has a high probability of destroying 90% of the targets than one than one which has a low probability

[1] the weight of effort specifies the percentage of aircraft which can be assigned to a particular target type, e.g. 40% of F-15s to SAM targets.

[2] phasing specifies the relative order target types should be attacked, e.g. all SAMs before bridges

of success but attacks 100% of the targets. The problem is identifying what percentage is possible and the targets in that sub-set. In addition, it is vital to avoid situations where many missions must be canceled or replanned because of small anomalies, such as a single target being missed.

3. Identifying the optimal break point at which the air campaign should switch from one target type to another. By finding the optimal break point it becomes possible to assign resources to attack high priority targets in temporally later target sets rather than using limited resources trying to destroy all the targets in an earlier target set the last few of which have relatively low value. Again this allows for more robust schedules which have a higher probability that they will achieve their overall aims.

The key to DEOSs ability to successfully solve problems in this domain is that it can generate schedules very quickly and be adaptable to changes in the task and the situation. There is no point in DEOS generating schedules for the next 12 hours when the schedule needs to change on a minute by minute basis. The core algorithm of the DEOS system is the "Squeakywheel" optimization (SWO) technique developed by Joslin and Clements [7]. The basic SWO algorithm has been modified to handle several new constraint types and these include probability distributions, probabilistic functions, temporal windows, resource limits and a limited set of precedence constraints. In addition a more expressive task description language [2, 1] has been integrated to allow the scheduler to better model the actual dynamics and activities in the domain. The algorithm has also been modified to allow it to identify optimal sub-sets of tasks from the task list and this technique is referred to as Penalty Box scheduling. These modification are generic and could be easily applied to problems in manufacturing, assembly, integration and test. Details of the task model and algorithm modifications are provided later in the paper.

Previous work [8] has addressed aspects of this problem but this approach differs in several important ways. The overall DEOS approach is to identify optimal resource assignments and where insufficient resources are available the best sub-set. The previous work [8] took an MDP approach to try and identify the best policy for a given target. This resulted in a solution in which the target may need to be attacked for several days consecutively and discussions for USAF pilots have shown that such a mission plan is usually a suicide one.[3] The DEOS approach is able to handle problems far larger and generate solutions in a few seconds as opposed to tens of minutes. In addition, the DEOS approach is able to handle a richer set of constraint types and optimization criteria (e.g., minimize makespan, maximize probability of damage and minimize attrition). Finally, the DEOS approach is able to handle the dynamic aspects of the problem (e.g. missed targets, pop-up targets) which the previous work cannot. This allows DEOS to develop schedules which are robust against certain types of change and minimize the knock on effects of changing missions on the fly. Current USAF planning systems use LP/IP solvers to generate mission schedules. The core SWO algorithm

[3] The enemy begin to expect the raids and hence the attrition rate becomes very high!!

has been compared with LP/IP solvers on several manufacturing problems and was found to out perform them in terms of the speed of solution and the quality of the solutions generated [4].

The paper is structured as follows, Firstly, it provides an overview of the ACP domain and the data used by the DEOS system. Secondly, it provides an overview of the task model and thirdly, it describes the basic scheduling algorithm and two extensions which allow it to identify optimal sub-sets. Fourthly, it provides details of the schedules generated and their evaluation by members of the USAF. Finally, it provides a summary of current progress and describes several additional techniques and ideas which will be explored.

1.1 Overview of the Target Scheduling Process

The weapon/aircraft pairing problem is a complicated one due to the many different trade-offs which are possible and the ability of the aircraft to be configured to suit different missions and different weapons loads. The actual weapon/aircraft pairing is based on a set of probabilities which take into account, probability of hitting the target, destroying the target[4] and the expected attrition rate of the aircraft against the target type. In theory any weapon/aircraft pairing could be sent against a target but it may have a very low chance of success. As described earlier some aircraft have a greater probability of success if additional assets are sent with them. For example, the expected aircraft attrition rate can be reduced by sending SEAD aircraft with the strike aircraft. However, this would mean that the SEAD aircraft could not be used as strike aircraft which may result in their being insufficient resources to attack a high value target later in the schedule. In addition to the constraints on individual targets and aircraft there are also further constraints relating to time and resource limits. The temporal constraints specify a window during which a target must be attacked, the window during which targets of a particular type can be attacked and the time delay between targets which are "connected" (e.g. the cooling towers of a power station must be attacked with 12 hours of the generator halls). The resource constraints specify the available quantities of aircraft and weapons (which can vary over time) and percentages limits on the number of aircraft which can assigned to a given target type[5] (e.g. 40% of missions against air defenses, 20% of missions against communication sites). These constraints are very problematic as the number of missions is not known in advance hence the scheduler needs to keep the percentages of different mission types in balance. The targets themselves are grouped into target sets (e.g. all bridges across the Thames) and these are then grouped into target systems (e.g. all railway centers in southern England). Unfortunately, the same target might be in two or more different target sets and hence has a higher "value" than the other targets in the same set. In addition, it may be the case that it is not necessary to attack all the targets in the set to achieve

[4] some weapons may be able to hit a target but not destroy it, e.g. an anti-tank missile can hit a building but it very unlikely to destroy it

[5] This is referred to as the Weight of Effort.

the overall aim. For example, if the aim is to stop the enemy forces crossing the river it may be possible to achieve this by destroying only 80% of the bridges. This makes target selection a very important aspect of the scheduling process. The scheduling process aims to find an optimal assignment of aircraft to targets which minimizes the probability of needing to restrike the target or cause collateral damage while also minimizing the risk to the assigned aircraft.

The problem is further complicated by the fact that aircraft can be reassigned to a different mission on the fly. For example, aircraft could be diverted to attack a pop-up target for which they are an optimal match. Alternatively, the aircraft may be a good match but the weapons they are carrying are not. This means the aircraft could be diverted to a base and reconfigured if time permits. Changing missions on the fly has potential knock on effects with later missions being postponed or reassigned due to longer than expected mission durations.

1.2 Mission Planning Data Models

The target matching problem is driven by a set of tables which provide details of the different aircraft, weapons, targets, support assets, etc. The primary information source is the target table and a section is provided in Table 1. This table show the type of mission, air superiority (AS) the hardness of the target and the risk associated with attacking the target [6]. Associated with each target type is a reward which defines the importance of the target to the human planners. Table 2 shows a section of the rewards table (these values were calculated through discussions with human planners and through the analysis of the schedules generated by DEOS).

Obj	Task	Lat	Long	Hardness	Risk	Target
AS	1	180804N	233902W	Hard	High	airfield
AS	3	172708N	223930W	Soft	Medium	radar-comms
...						

Table 1. ACP Target Table

The mission type from Table 1 identifies the class of aircraft which could be sent against the target. Table 3 shows the mapping of aircraft to mission type and shows that the same aircraft can be used in for many different missions.

One of the optimization criteria for this problem is minimize risk and DEOS tries to identify aircraft which have a low risk against a selected type of target. The expected risk to an aircraft is calculated by summing the total probability that the aircraft will be shot down either to or from the target[7] One key decision

[6] Not shown is the time window during which the target needs to be attacked, e.g. D+5, D+10, etc

[7] SAM batteries have a threat radius which has a known probability of detection based on the distance the aircraft is from the center of the radius.

Mission	Reward
mil activity	45
SAM site	90
SSM site	100
C3	35
command HQ	60
...	

Table 2. Target Reward Table

DEOS needs to make is to whether or not to use SEAD protection to reduce the risk to the attacking aircraft. As pointed out earlier this may have the side effect of making another strike mission late.

Once a potential aircraft has been selected it must be checked to ensure that it can carry appropriate weapon load for the target. The probability of destruction is noted in terms of a single weapon and the current USAF doctrine is that the plane carries enough weapons to give a 90% or better chance of destroying the target. There is no guarantee that a specified weapon load will destroy the target as they could all miss. Additional tables provides details of the probability of weapon hitting the target, provide data on air to air refueling times, aircraft speed and range, turn round times, etc. Full details of the aircraft/weapon pairing algorithm are given later in the paper.

Mission	Aircraft
Counter Air	F15E, F117, F16C, F16CLN, F14A, F14B, F14D, FA18A, FA18C, FA18D, AV8B, B52H, B1B
SEAD	F16CJ, EA6B, FA18C
Def Counter Air	F14A, F14B, F14D, F16C
Interdiction	F15E, F117, F16C, F16CLN, B52H, B1B, F14A, F14B, F14D, FA18A, FA18C, FA18D, AV8B
Close Air Support	A10, AV8B, F16C, F15E, FA18A, FA18C, FA18D
Strategic Attack	F117, B52H, B1B

Table 3. Aircraft and Mission Mapping Table

Each mission is modeled using the PRFER mission task model [5] that defines a natural breakdown of a mission into its constituent parts or sub-blocks.

- **P**lan: Time taken for the pilot to plan the mission. Once a plan has been identified it is inserted in the slot for other workflow tasks to examine and check.
- **R**eady: Time taken to prepare the plane for the mission

– **F**ly: Time taken to get to the mission objective [8]
– **E**xecute: Time taken to execute the mission, e.g. drop weapons, unload food pallets, etc.
– **R**econstitute: Time taken to turn the aircraft round once it has returned to base.

The PRFER model allows a tasking agent to create a better model of the processing the task needs and to better understand how to allocate resources, identify tradeoffs, asses changes and modify the associated task list. The sub-blocks are allowed to "breath" as changes in the domain are reflected as changes in one or more of the sub-blocks. For example, if the aircraft chosen for the mission develops a failure during its ready time then the "Ready" sub-task will expand and accommodate the extra time. The PRFER model allows DEOS to quickly identify the impact of changes, propose potential changes to the mission tasking and inform the planners of new deadlines and constraints (e.g. the planes now on hold need refueling in the next 30 minutes).

2 Resource Allocation Algorithm

The basic concept behind DEOS is to generate schedules quickly and to update them on the fly as new requirements and changes occur in the domain. The core SWO algorithm uses a priority queue to determine the order in which tasks should be released to a greedy scheduling algorithm. This identifies the best aircraft/weapon for a given task from those available. Tasks later in the priority queue have a smaller choice of resources due to earlier commitments. The order of the priority queue is determined by how difficult the task is to deal with that is, the higher the task is in the queue the harder it is to handle it correctly. It does not require an external priority to be identified by the user. Once a schedule has been generated it is analyzed to identify which tasks were handled badly (e.g., a task was completed after its deadline, or assigned to a high attrition aircraft). Any task that "squeaks" (i.e., was handled badly) is given a "blame score" and is promoted in the priority queue, with the distance it is promoted determined by the extent of the problem. This new priority queue is then used to generate another schedule that is analyzed for problems. This process continues until no significant improvement in the schedule is noted over several iterations. SWO is extremely fast with each cycle of generate, analyze, and re-prioritize taking only a few seconds, even for large problems.

One of the key issues in this domain was to generate schedules which balanced a number of potentially conflicting factors. For example, the planners wanted all 2500 targets attacked in the shortest time, with minimum attrition and minimum risk of collateral damage. However, to guarantee that each target was attacked with minimum risk would require all missions to be flown by F-117s and that would result in very long schedules. A sample schedule was generated which used

[8] This can be replaced by a "drive" or "sail" block for operations using land or sea transport

only the best target/aircraft pairing and it had a makespan in excess of six days. Using the DEOS approach the schedule was reduced in length to just under two days with a less than 1% reduction in overall schedule quality.

To address these potential conflicts a series of functions were developed which investigated the different aspects of the problem, e.g. aircraft attrition, probabilities of hitting and destroying the target, numbers of weapons needed, number of aircraft needed, support assets, number of sorties, etc. It was identified that the key elements of evaluation were the probability that the target would be attacked successfully and that the attacking aircraft would have a low attrition rate. This allowed two main functions to be identified [9]. Function 1 describes the probability that a target will be destroyed given a specified number of weapons `W` and the probabilities of hit and kill (P_h P_k) respectively for a single weapon. Function 2 describes the expected attrition rate for `n` aircraft when attacking with `N` total aircraft.

Function 1:

$$P_{kill}(W, P_h, P_k) =$$
$$P_h^W \sum_{n=1}^{w} \binom{w}{h} \left(\frac{1-P_h}{P_h}\right)^{w-n} \left(1-(1-P_k)^N\right)$$

Function 2:

$$P_{attrition}(N, n, P_a) = \binom{N}{n} (1-P_A)^{N-n} P_A^n$$

The DEOS uses these formulas to evaluate different combinations of weapons and aircraft for a given target type, trying to identify the best possible match. However, it may be the case that the required aircraft/weapon pairing may be unavailable in the desired time interval (e.g., between 0900 hrs and 1100 hrs all F-16s may be assigned to other missions). DEOS may decide to use a second option (i.e., a different aircraft and/or weapon) and will cycle through the different options until an assignment of aircraft/weapons to the target can be made [10]. In addition, DEOS may add in a SEAD sortie to off set a high expected attrition rate. After an assignment has been made it may be the case that it is a poor one (e.g., high attrition rate, low probability of success) and this is dealt with in the next cycle of algorithm when the generated schedule is analyzed and poor assignments identified.

During the development of the algorithm it was identified that in many cases the number of targets greatly exceeded the available resources. In addition, it was also identified that some of the time constraints provided by the human schedulers were leading to less than optimal schedules. Details of the modifications to the basic algorithm are provided in the following sections.

---

[9] Other support functions were developed but are not discussed in this paper

[10] By default the targeting database provided 5 options but the 4th and 5th usually had a low probability of success

2.1 Penalty Box Scheduling

The aim of penalty box scheduling is to identify a sub-set of tasks which can be resourced effectively and avoid the problem of generating low quality schedules which resource all tasks. For example, human planners may be happier striking 90% of the targets with high probability of success rather than 100% of the targets with a much lower probability of success (i.e., the planners wanted robust solutions which has a higher probability of success). The problem is finding what percentage can be assigned and which tasks to select. Penalty box scheduling extends the SWO algorithm by viewing the inability to assign a task within its specified time window as a high priority problem (i.e., a large squeak). Instead of placing the task at a point later in the schedule the task is put in the penalty box[11] for a single cycle of the algorithm. The penalty tasks are assigned a high blame value and their position in the priority queue altered. The blame value also takes into account the potential reward for striking the target and the external priority assigned by the user to the target set. At the end of the scheduling process[12] those tasks in the penalty box are left unassigned. This extension proved highly efficient (i.e., there was a negligible slow down in the speed of solution) at identifying sub-sets of tasks and provided the human schedulers with more robust solutions to the targeting problem. After scheduling was complete the human planners were able to provide feedback on which tasks left in the penalty box needed to be resourced. They could then compare the resulting schedule with the optimal one and measure (i.e., number of missions, sorties, expected attrition rate, etc) the drop in the overall schedule quality.

2.2 Temporal Phase Transition

Missions are specified with time windows during which the mission must be accomplished. However, these associated time windows tend to be arbitrary and estimates by the human planners. Rather than use the time constraints as invariable, DEOS was allowed to relax them and attempt to identify the point at which to switch from one mission type to another. For example, attacking SAM sites should be completed first (for the next 6 hours) and then attacks against power stations for the next 6 hours. Their division may mean that fairly low priority SAM missions can be handled whereas only the highest priority power station missions can be assigned. A better schedule may be to stop SAM missions after 4 hours and give the additional 2 hours to the power station missions. The selection of suitable subsets needs to be weighted against the flexibility built into the schedule by allocating maximal windows. For example, more tasks might be resourced within a window at the expense of making the schedule more brittle.

The temporal phase transition problem was investigated through two different methods. The first method involves a variation of the penalty box scheduling

[11] This is a term connected with sports where a player committing an offense is placed in the penalty box for a specified period.

[12] DEOS keeps track of the best schedule found so far and its associated penalty box entries.

algorithm in which pointers are maintained to the last task of the temporally earlier set and to the earliest task of the later set respectively. It always the case that no SAM can be placed after any power mission. For example, if a SAM mission cannot be scheduled before the earliest power mission then it is sent to the penalty box for a cycle. Alternatively, if a power mission can be scheduled after all SAM missions but before the current earliest power mission then it can be added and the pointers updated. This relies on the ability of the critiquing phase of the SWO algorithm to apportion blame appropriately to move the missions in the penalty box the required distance in the priority queue. The second method involves rippling all the power missions to the right to fit in a new SAM mission. Any power missions already in the schedule keeps their assignment (i.e. a F-16) but are moved later in time (i.e., they do not have to accept a lower quality assignment). If the tasks cannot be rippled right then the new task is assigned to the penalty box. This relies on the construction phase of SWO algorithm being able to reconstruct new partial assignments on the fly. By having already assigned power tasks keep their assignments (or be assigned one no worse (i.e., swap the F-16 or a F-15) it keeps the problem tractable. The analysis of the schedule showed that on problem sizes up to 2000 tasks it was better to use the shuffle approach and for problems greater than 2000 the pointer approach was marginally better.

3 Results

Figure 1 shows the performance of DEOS on an example test set of 700 targets and 150 aircraft. The optimization criteria included low attrition rate, high probability of success and a minimal makespan. The best schedule identified completes all 700 targets in 47 hours with an expected loss rate of less than 1%. To date the DEOS results are the best for these problem and easily surpass those developed by current USAF mission planners. Figure 1 shows that the addition of penalty box scheduling and phase transition components does not effect the overall performance of the system. DEOS very quickly settles in an appropriate region of the search space and spends many iterations trying to improve on a reasonably good schedule. DEOS is trying to identify trade-offs between the different optimization criteria and Table 4 shows a typical example. Between iteration 2 and 3 the `raw` score[13] increased by less than 1% but the `analysis` score[14] increased by nearly 25% due to the schedule being a lot shorter.

The example above also shows that DEOS was able to find the best sub-set of targets from those specified (e.g., 667 out of 700 were successfully tasked). The DEOS schedules allow USAF planners to identify robust solutions and the incremental costs (e.g., additional planes, sorties, attrition) necessary to attack all targets. For each target DEOS identifies an appropriate number of aircraft, weapon load and timing information. In some cases the assigned aircraft/weapon

[13] This is the summation of the number of targets attacked, probability of success, number of missions and sorties

[14] This is the raw score divided by the makespan in minutes

Iteration	Targets Assigned	Raw Score	Analysis
2	667	667075	15744
3	667	669873	20219

Table 4. Target set vs Makespan Trade-off

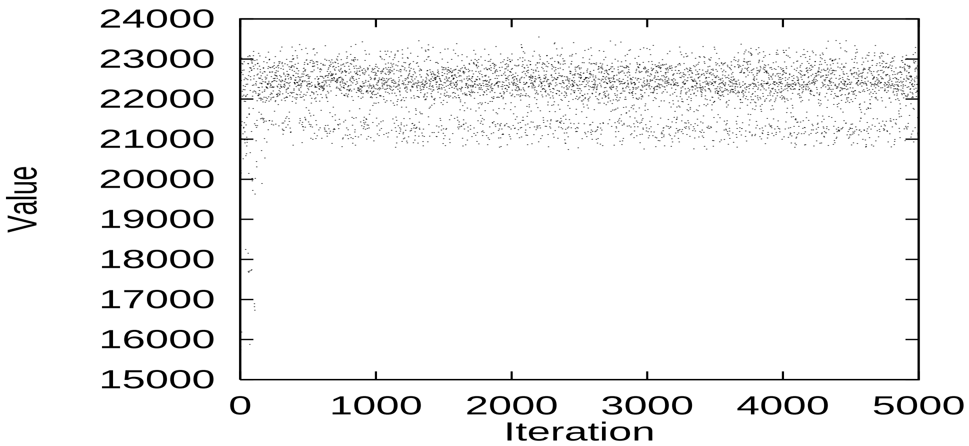

Fig. 1. Air Mission Planning Results

is a less than optimal match. It is often the case that to obtain a good overall schedule some tasks need to be handled badly (i.e., they need to be sacrificed). It is possible to handle the sacrificed tasks better but only at the expense of making the overall schedule worse. The presence of "sacrifice tasks" usually indicates that additional resources of a particular class are needed. The system was evaluated by subject matter experts (SME) from the USAF. The aim was to show that the SME's view of schedule quality and that of DEOS were correlated. The SMEs were given pairs of schedules whose difference in quality narrowed gradually and were asked to choose the better schedule. In all cases the view of the SME and DEOS was correlated. After six iterations the SMEs were unable to make an informed decision over which schedule was better.

4 Summary and Further Work

This paper has presented a description of the DEOS scheduling system, its scheduling algorithm and its application to the mission scheduling problem. DEOS allows for the explicit analysis of trade-offs in resource allocation, dynamic update of on going schedules, on the fly task addition and for focussed impact analysis and repair. To date the system has been applied to large scale ACP problems (i.e. 2500 targets and 200 aircraft over a 5 day period) and was successfully demonstrated as part of the USAFs Effects Based Operations project at the end of 2000. The techniques are generalizable to other domains in which there are flexible time constraints and the "penalty box" techniques are applicable to problems where there is phasing between different groups of tasks. For example, in manufacturing domains schedulers are often faced with the problem of switching production from one type to another to improve overall productivity. Several improvements will be made to DEOS and these include adversarial planning in which the schedule will propose robust solutions to potential enemy responses. The interface will be improved to allow easier interaction and specification of policies and preferences. The results from the ACP domain and other non-probabilistic manufacturing domains show a distinct grouping of schedule quality as shown in Figure 1. These groups represent classes of solutions (rather than point solutions) that have particular attributes and values. DEOS will be modified to automatically identify these discontinuities in the solution space and alert the planners.

References

1. Berry, P.M. and Drabble, B: SWIM: An AI-based System for Workflow Enabled Reactive Control, in the Proceedings of the Workshop on Workflow and Process Management held as part of the International Joint Conference on Artificial Intelligence (IJCAI-99), (eds, B, Drabble and M. Ibrahim), IJCAI Inc, August, 1999.
2. Berry, P.M. and Drabble, B.,: The AIM Process Modeling Methodology, AI Center, SRI International, Technical Report, 1789, Menlo Park, CA, 1999.
3. Chien, S., Knight, R., Stechert, A., Sherwood, R., and Rabideau, G.,: Integrated Planning and Execution for Autonomous Spacecraft, in the proceedings of the IEE Aerospace Conference (IAC), Aspen, CO, March 1999.
4. Clements, D., Crawford, L., Joslin, D., Nemhauser, G., Puttlitz M. and Savelsbergh, M.: Heuristic Optimization: A hybrid AI/OR approach, in the proceedings of the Workshop on Industrial Constraint-Directed Scheduling, 1997. (Held in conjunction with CP'97, Schloss Hagenberg, Austria.).
5. Drabble, B.: Task Decomposition Support to Reactive Scheduling, in the proceedings of the 5th European Conference on Planning (ECP-99), Springer Verlag Press, New York, NY, USA, September, 1999.
6. Fox, M.S.: ISIS: A Retrospective, in Intelligent Scheduling, (Zweben. M. and Fox, M.S.), 1994. Publisher Morgan Kaufmann, Palo Alto, CA, 94303, USA, pp3-28.
7. Joslin, D.E. and Clements, D.P.: Squeakywheel Optimization, in the proceedings of the Fifteenth National Conference on Artificial Intelligence, Madison, WI, AAAI Press, Menlo Park, CA, USA 1998.
8. Meuleau, N., Hauskrecht, M., Kim K., Peshkin. L., Pack Kaelbling. L., Dean. T., and Boutilier., C.: Solving Very Large Weakly Coupled Markov Decision Processes, in the Proceedings of the Fifteenth National Conference on Artificial Intelligence, AAAI Press/MIT Press, MIT, Cambridge, MA 02142, USA, July, 1998, pp165-172.
9. Myers, K.L..: A Continuous Planning and Execution Framework, AI Magazine, Vol 20(4), AAAI Press, Menlo Park, CA, 1999.
10. Sadeh, N.: Micro-Opportunistic Scheduling: The Micro-Boss Factory Scheduler, in Intelligent Scheduling, (Zweben. M. and Fox, M.S.), 1994. Publisher Morgan Kaufmann, Palo Alto, California, 94303, USA, pp99-136.
11. Smith, S.F.: OPIS: A Methodology and Architecture for Reactive Scheduling, in Intelligent Scheduling, (Zweben. M. and Fox, M.S.), 1994. Publisher Morgan Kaufmann, Palo Alto, CA, 94303, USA, pp29-66.