P-MEP: Parallel More Expressive Planner

Javier Sanchez (*), Minh Tang & Amol D. Mali
Electrical Engineering & Computer Science,
University of Wisconsin, Milwaukee, WI 53211,
(*) EDESA S.A, Cra 18 86A-14, Bogota, Colombia, { javier, minhtang, mali }@uwm.edu

P-MEP [10] is a forward state-space planner that performs weighted A* style search. It allows a user to choose the heuristic to be used and the weight in weighted A*. It has relevance analysis as a preprocessing technique to control search. P-MEP uses the notions of referenced and updated variables to detect equivalent states to control search. The key ideas in P-MEP are the use of mutual exclusion relations (mutexes) in the computation of relaxed plans and the use of intervals of relaxed values. The notion of relaxed intervals in P-MEP is inspired by relaxations in Sapa and Metric-FF. The relaxed intervals allow P-MEP to handle expressions containing +, -, /, *, exponentiation, $=, <, >, \leq, \geq, \wedge, \vee, \neg, \rightarrow, \forall$, and $\exists$. Relaxed interval of a variable contains the minimum and maximum relaxed values of the variable. Relaxed intervals are useful in several ways. They allow P-MEP to handle decrease effects of actions and numerical preconditions in the computation of relaxed plans. P-MEP uses the relaxed intervals to check if numerical preconditions are achievable in a relaxed fashion. Relaxed intervals also allow P-MEP to handle linear and non-linear expressions in the goal of a planning problem. The relaxed intervals also allow an easy detection of whether a numerical goal or numerical subgoal is achievable in a relaxed fashion.

P-MEP creates all ground instances of operators before search begins. If the domain description does not specify operator durations, P-MEP assumes that they are all unity and treats the domain as a temporal domain.

World State: P-MEP treats propositions and ground predicates as numerical variables with domain $\{0, 1\}$. A world state S in a node N in the search tree of P-MEP is the tuple $< V, a, t >$, where t is time stamp of S, V is the set of numerical variable-value pairs, and a is the action applied to generate N. Time stamp of a world state S is the earliest time at which an action can be applied in S. P-MEP applies actions so that they start at the earliest possible times.

Applicable Actions: The conditions for applicability of an action o' in a node n with world state S are: (i) all preconditions of o' that need to be true at its start or end points must be satisfied by V of S, (ii) all preconditions of o' that need to be true over its entire interval must also be satisfied by V of S, and (iii) effects of o' do not change the value of any boolean or numerical variable in the preconditions or effects of any action that o' overlaps with and vice versa.

Statically Mutex Actions: For each action o', P-MEP computes the set of variables updated by the action and the set of variables that are referenced by the action. These sets are denoted by $U(o')$ and $R(o')$ respectively. These sets contain predicates and discrete variables. As an example, let the preconditions of o' be a, b and $c > 200, e < 100$, where a and b are propositions and c, e are discrete variables. Let the effects of this action be $\neg a, d, c = (c + e - 50)$, where d is a proposition. In this case, $R(o') = \{a, b, c, e\}$ and $U(o') = \{d, a, c\}$. In general, any variable that appears in preconditions or effects of an action and which is not updated by the action is a referenced variable for that action. Two actions o_i and o_j are statically mutually exclusive in P-MEP if one or more of the following three conditions are satisfied: (i) $(R(o_i) \cap U(o_j)) \neq \phi$, (ii) $(R(o_j) \cap U(o_i)) \neq \phi$, (iii) $(U(o_i) \cap U(o_j)) \neq \phi$. Statically mutually exclusive actions are permanently mutually exclusive and cannot overlap.

Equivalent States: R is the set of variables referenced by one or more actions and U is the set of variables updated by one or more actions. Two states s and s' are equivalent if the values of all variables in $R \cap U$ are same in the V component of s and s'. This definition of equivalent states allows P-MEP to control search by not visiting multiple world states that differ only in the value of variables from $U - R$. The variables in $U - R$ do not affect the applicability of actions. Consider the variable total-fuel-consumed denoting the total fuel consumed by a partial plan, in transportation logistics domain. This variable is not relevant to achieving any precondition of any action. This variable belongs to U but not to R. This variable can have infinite non-negative values since infinite flights are possible. By not considering such variables in the state equivalence test, P-MEP controls the size of its search tree.

Search Algorithm: P-MEP conducts forward state-space search in weighted A* style. The weighted variant of A* uses the following path cost equation $f(n) = (1 - w) * g(n) + w * h(n), 0 \leq w \leq 1$, where $g(n)$ represents the cost of the path from the root node to node n, and the $h(n)$ represents the estimate of the cost of the cheapest path from n to goal. In P-MEP, the nodes in fringe are sorted according to value of the f function. The node with lowest $f()$ value is expanded first. If multiple nodes have the same value of f, then the node with lowest depth is expanded first. If nodes with the lowest value of f have the same depth, then the node that is generated earlier is expanded first. A node is generated by applying only one action. Since multiple actions may have the same starting time, concurrency is possible. An action o' can start during the interval of other actions that are not statically mutex with o', making concurrency possible. P-MEP terminates when there is a node n such that every subgoal is true in the V component of the world state in n.

Relaxed planning graph (RPG): The notion of RPG was introduced in FF planner [6]. We denote the goal of a planning problem by G in the rest of the paper. A subgoal from G is an expression from G. An RPG is constructed by FF assuming that the delete effect lists of actions are empty. The notion of proposition level is replaced by the notion of variable level, in order to construct an RPG for more expressive domains. i th action level occurs between i th variable level and $i + 1$ the variable level. P-MEP constructs a serial relaxed planning graph. P-MEP stores an interval bounded by maximum and minimum relaxed values for each variable in each variable level. A variable level is a set of $< v, [min(v), max(v)] >$ tuples, where $min(v)$ and $max(v)$ are minimum and maximum relaxed values of variable v. The size of an interval is monotonically increasing. The interval for a variable v' in i th variable level is obtained by updating its interval in the $i - 1$ the variable level with the effects of the action in the $i - 1$ th action level. For example, let the value of variable v_1 in the world state of a node n be 4. Then the interval for v_1 in the first variable level in the RPG at n is [4,4]. If an action increasing v_1 by 10 is included in the first action level of the RPG, then the interval for v_1 in second variable level is [4,14]. If an action decreasing v_1 by 20 is then included in the second action level of RPG, the interval for v_1 in the third variable level is [-16,14]. If an action assigning 5 to v_1 is then included in the next action level of RPG, the interval for v_1 in the fourth variable level is still [-16,14], since 5 is in the interval [-16,14]. The intervals for variables make it easy to compute relaxed intervals for expressions and check if the expressions are satisfied in a relaxed fashion in a variable level. For example, let the relaxed intervals for variables v_1, v_2 be [-3,5] and [-4,8]. The interval of $(v_1 + v_2)$ is [-7, 13]. The intervals for $v_1 * v_2$, $\frac{v_1}{v_2}$, and $v_1 - v_2$ are found in a similar fashion. The intervals of complex arithmetic expressions are found using intervals of individual variables and operator precedence. For example, the relaxed interval for $v_1 * v_2 * v_3 * v_4$ is found from relaxed intervals of $v_1 * v_2 * v_3$ and v_4. The interval of $v_1 * v_2 * v_3$ is found from the intervals of $v_1 * v_2$ and v_3. The intervals for variables and expressions can be considered as the intervals of relaxed values. This is because though P-MEP considers add,delete effects, increase, assign and decrease effects in computing intervals, it ignores the interactions between actions. So some of the values in the intervals may be impossible to achieve.

Intervals of expressions in preconditions or goal are found only to test if preconditions or goal are achieved in a relaxed fashion. Intervals for expressions make it easy to check if actions are applicable in RPG and if goal is true in RPG. For example, the precondition or subgoal $(v_1 + v_2) = 50$ is true in RPG if 50 lies in the interval of $v_1 + v_2$. Similarly, the expression/subgoal $v_1 < v_2$ is satisfied in a variable level in the RPG if $min(v_1) < max(v_2)$ is satisfied in the variable level.

P-MEP constructs RPG for a node by applying actions in forward direction and by computing action and variable levels, until the intervals of variables satisfy all expressions in the goal in some variable level or no variable's interval changes, whichever occurs earlier. If some subgoal is not achieved in the RPG of a node n, P-MEP sets $h(n)$ to ∞ and keeps the node in the priority queue. n is be expanded after all states with finite $h()$ values are expanded.

Relaxed plan: Relaxed plans are used to compute $h()$ values for nodes by P-MEP, like Metric-FF [7], Sapa [2]. Relaxed plan for a node n is found by P-MEP in two phases. In first phase, it removes irrelevant actions from the RPG of n. This removal leaves a subgraph of RPG with gaps (some action levels are empty). In the second phase, P-MEP converts this subgraph into a relaxed temporal plan by pushing actions back to the earliest possible time, ensuring that statically mutex actions do not overlap. P-MEP considers action durations only in the second phase.

The relaxed plan found by phase 1 is serial. It

PLN	PL	NV	TD	NPG	NGPG	CE	Q	DPG
Sapa	3	Y	Y					
LPG	3	Y	Y					
MIPS	1, 2, 3	Y	Y		Y		Y	
TP4	3	Y	Y					
MFF	1, 2	Y		Y	Y	Y	Y	Y
FF	1				Y	Y	Y	Y
VHPOP	1, 3	Y			Y	Y	Y	Y
P-MEP	1, 2, 3	Y	Y	Y	Y	Y	Y	Y

Figure 1: Expressiveness features handled by various planners from 2002 planning competition. Y: Yes (handled).

is parallelized in phase 2 because temporal planning problems generally involve makespan minimization. Makespan of parallelized relaxed plan of node n can be a better estimate of the makespan of the optimal plan that achieves the goal from node n, than the makespan of serial relaxed plan. The estimates of the makespan of optimal plan from n can be better if statically mutex actions do not overlap in the relaxed plan at n. Hence overlap of statically mutex actions is avoided in the parallel relaxed plan.

Supported Domain Features: The domain features supported by P-MEP and seven other planners that participated in the international planning competition in 2002 are shown in Table 1. TP4 and VHPOP are described in [5] and [8] respectively. The acronyms in this table have the following meanings: PLN: Planner, MFF: Metric-FF, PL: PDDL level, NV: Numeric variables, TD: Time Durations, NPG: Numerical preconditions and goal, NGPG: Negated preconditions and goal, CE: Conditional Effects, Q: Quantifiers, DPG: Disjunctive preconditions and Goal. PDDL is planning domain description language. PDDL 2.1 level 1 includes STRIPS and ADL. PDDL 2.1 level 2 is an augmentation of PDDL 2.1 level 1 with numeric variables. PDDL 2.1 level 3 is an augmentation of PDDL 2.1 level 2 with time. PDDL levels partially/fully handled by various planners are also shown in Table 1. P-MEP is the only planner that handles all domain features in Table 1. The most recent version of MIPS does handle ADL.

Relevance Analysis: This is used as a preprocessing technique to reduce the number of actions used in search. This technique is similar to relevance analysis in [9]. P-MEP gives a user an option to use relevance analysis. P-MEP constructs an extended and serial relaxed planning graph (ESRPG) for root node by applying actions in forward direction, as a part of relevance analysis. The RPG is extended because its growth may be continued even after all subgoals are achieved in a relaxed fashion in some variable level. P-MEP does not check for the achievement of subgoals in the variable levels when it constructs the ESRPG. The construction of ESRPG stops when no new action is applicable and then the set of actions in various action levels in the ESRPG is returned as the relevant actions' set.

Heuristics: P-MEP allows user to choose a heuristic from the following four heuristics: Cost, Makespan, Sum duration and Actions. The heuristics are not new, but the actual heuristic values and plans differ due to different method of computing relaxed plans in P-MEP.

[1] A. Blum and M. Furst, Fast Planning through planning graph analysis, . Artificial Intelligence, Vol.90(1-2), 1997, pp. 281-300.

[2] M. Do and S. Kambhampati, SAPA: A multi-objective metric temporal planner, JAIR 20, 2003, pp. 155-194.

[3] S. Edelkamp, Taming numbers and durations in the model checking integrated planning system, JAIR 20, 2003, pp. 195-238.

[4] A. Gerevini, A. Saetti, and I. Serina, Planning through stochastic local search and temporal action graphs in LPG, JAIR 20, 2003, pp. 239-290.

[5] P. Haslum and H. Geffner, Heuristic planning with time and resources, Proceedings of the European Conference on Planning, 2001.

[6] J. Hoffmann and B. Nebel, FF: The FF Planning System: Fast plan generation through heuristic search, Journal of Artificial Intelligence Research, Vol. 14, 2001.

[7] J. Hoffmann, The Metric-FF Planning System: Translating "Ignoring Delete Lists" to Numeric state variables, JAIR 20, 2003, pp. 291-341.

[8] H. Younes and R. Simmons, VHPOP: Versatile Heuristic Partial-order Planning, JAIR 20, 2003, pp. 405-430.

[9] J. Hoffmann and B. Nebel, RIFO Revisited: Detecting relaxed irrelevance, Proceedings of ECP, 2001.

[10] J. Sanchez, Integrating efficiency and expressiveness in planning, M.S thesis, Computer science, University of Wisconsin, Milwaukee, April 2003.