Representation Changes for Planning with Exponential Utility Functions

Sven Koenig
Georgia Institute of Technology, College of Computing
Atlanta, Georgia 30332-0280
skoenig@cc.gatech.edu

Abstract

Representation changes are often used to make planning more efficient. We, on the other hand, show that they can also be used to extend the functionality of existing planners. We study probabilistic planning tasks that require one to maximize the expected total utility for a given convex exponential utility function. Delivery tasks with immediate soft deadlines, for example, fit this description. We introduce the additive and multiplicative planning-task transformations, that both perform fast representation changes that transform planning tasks with exponential utility functions to planning tasks that standard search and planning methods from artificial intelligence can solve. Advantages of these representation changes include that they can be performed on a variety of planning-task representations without changing their kind or size. They make planning with exponential utility functions as fast as planning for traditional preference models and can easily be integrated into existing planning architectures. Finally, they make it possible to participate in performance improvements achieved by other researchers in the currently very active field of artificial intelligence planning.

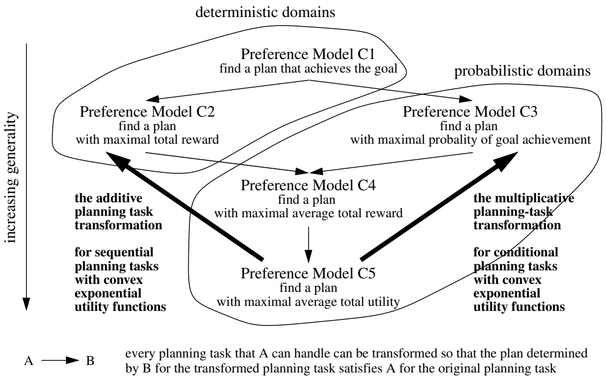

Figure 1: Preference Models for Planning Tasks

1 Introduction

Planning methods have to decide which of the plans that solve a given planning task they consider best. The preference model of most probabilistic search and planning methods from artificial intelligence is to find plans with maximal probability of goal achievement or plans with minimal average plan-execution cost. Utility theory, on the other hand, suggests a more general preference model, namely, to find plans with maximal average total utility, where the utility is a strictly monotonically decreasing function of the plan-execution cost. Maximizing average total utility and minimizing average plan-execution cost result in the same decisions if either the domain is deterministic or the utility function is linear, but these properties rarely hold. Thus, we study how to find plans with maximal average total utility efficiently for nonlinear utility functions in probabilistic domains. We focus on one particular class of nonlinear utility functions, namely convex exponential utility functions, because they allows us to generalize the preference model of search and planning methods from artificial intelligence without giving up efficiency. These utility functions are, for example, well suited for expressing planning tasks with immediate soft deadlines, including delivery tasks. We develop efficient representation changes that transform planning tasks with convex exponential utility functions to planning tasks that search and planning methods from artificial intelligence can solve. Thus, while representation changes are often used to make planning more efficient, we use them to extend the functionality of existing planners. Our representation changes are simple context-insensitive transformations that can be performed locally on various representations of planning tasks. The additive planning-task transformation applies to planning tasks that can be solved with sequential plans (more precisely: planning tasks whose actions have deterministic outcomes but whose rewards can be nondeterministic) and transforms them to deterministic planning tasks. The original planning task can then be solved by finding a plan with minimal plan-execution cost for the transformed planning task. The multiplicative planning-task transformation applies to planning tasks that can be solved with conditional plans and transforms them to other probabilistic planning tasks. The original planning task can then be solved by finding a plan with maximal probability of goal achievement or a plan with minimal average plan-execution cost for the transformed planning task. Both transformations are summarized in Figure 1. We illustrate them using path planning for delivery tasks in the presence of blockages and immediate soft deadlines.

2 Traditional Approaches

Plans in probabilistic domains can have more than one chronicle, where a *chronicle* is a specification of the state of the world over time, representing one possible course of execution of the plan. We make the following assumptions: Although the agent cannot determine a priori which chronicle results from the execution of a plan, it is often realistic to assume that the agent is able to determine the probabilities with which the chronicles occur and their total rewards. The probability of a chronicle is the product of the probabilities of its action outcomes. This assumes probabilistic independence. The total reward of a chronicle is the sum of the rewards of the actions that it contains. This assumes an additive value function. Larger rewards are preferred over smaller rewards.

Given these assumptions, not all planning methods share the same planning objective since they use different preference models to compare plans. In the following, we discuss common preference models and relate them to the preference model studied in this paper. **Preference Model C1:** Traditionally, planners have been used in deterministic domains with the objective to find plans that achieve the goal. **Preference Model C2:** To make the preference model richer, planners then began to associate rewards (execution costs) with actions and preferred plans with maximal total reward (minimal plan-execution cost). A plan with maximal total reward achieves the goal, for example, if the goal can be achieved, all actions have negative rewards, and plan execution stops only in goal states. We are interested in generalizing these preference models to probabilistic domains. Several generalizations are possible. **Preference Model C3:** When the goal cannot be achieved for sure, or it takes too long to find plans that achieve the goal, plans are often preferred that maximize the probability of goal achievement. In particular, if the execution of plan ps leads with probability p_i to chronicle i and the chronicles $i \in I$ achieve the goal, then its *probability of goal achievement* is $\sum_{i \in I} p_i$. **Preference Model C4:** When the goal can be achieved for sure, plans are often preferred that maximize the average total reward. In particular, if the execution of plan ps leads with probability p_i to a chronicle with total reward r_i, then its *average total reward* is $er(ps) := \sum_i [p_i r_i]$ (er = expected reward). **Preference Model C5:** In this paper, we prefer plans with maximal average total utility, where the utility is a strictly monotonically increasing and twice continuously differentiable function that maps rewards r to the resulting real-valued utility $u(r)$. In particular, if the execution of plan ps leads with probability p_i to a chronicle with total reward r_i, then its *average total utility* is $eu(ps) := \sum_i [p_i u(r_i)]$ (eu = expected utility). Instead of using this preference model directly, we often use the following equivalent preference model. We prefer plans with maximal certainty equivalent, where the *certainty equivalent* of plan ps is $ce(ps) := u^{-1}(eu(ps))$ (ce = certainty equivalent). This is equivalent to preferring plans with maximal average total utility since a larger certainty equivalent implies a larger average total utility, and vice versa.

Figure 1 depicts the relationships among the five preference models: maximizing average total utility is more general than the other preference models. Utility theory [Bernoulli, 1738; von Neumann and Morgenstern, 1947] has studied how to use utility functions to express preference models. Unfortunately, it is a purely descriptive theory that specifies only what optimal plans are, but not how they can be obtained other than by enumerating every chronicle of every possible plan. Operations research has picked up on the results from utility theory and uses dynamic programming methods to find plans with maximal average total utility [Marcus *et al.*, 1997]. These methods do not utilize available domain knowledge. Artificial intelligence has investigated knowledge-based search and planning methods that scale up to larger domains, but traditionally do not determine plans with maximal average total utility. In this paper, we therefore extend the range of planning methods that find plans with maximal average total utility from methods from operations research to search and planning methods from artificial intelligence. We focus on convex exponential utility functions. Exponential utility functions are perhaps the most often used utility functions in utility theory [Watson and Buede, 1987] and specialized assessment procedures are available that make it easy to elicit them from decision makers [Farquhar, 1984; Farquhar and Nakamura, 1988]. Convex exponential utility functions are utility functions of the form $u(r) = \gamma^r$ with $\gamma > 1$. They can, for example, model (either exactly or approximately) immediate soft deadlines of different softness [Koenig and Simmons, 1996]. *Immediate deadlines* coincide with the time at which the execution of the plan begins. *Soft deadlines* are those whose utility does not drop to zero immediately after the deadline has passed but rather declines slowly. Consider, for example, delivery tasks. Often, delivery requests are not made in advance, the goods are needed right away, and the utility of the deliveries declines over time, for example because the deliveries can become obsolete or the delivery items lose their value over time (coffee, for example, gets cold). An example of a utility function that represents an immediate soft deadline is shown in Figure 2. It is the convex exponential utility function $u(r) = (2^{1/300})^r$, where r is the negative travel time of the robot, measured in seconds. In this particular case, the utility of the delivery halves every five minutes.

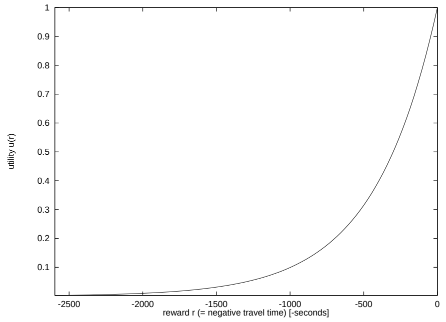

Figure 2: Convex Exponential Utility Function

3 Planning with Exponential Utility Functions

In the following, we show how to find plans with maximal average total utility (preference model C5) efficiently for convex exponential utility functions. We assume that exactly one resource (such as time, energy, or money) is limited. The reward of an action is the negative of the amount of the resource that it consumes, implying an additive value function. Plan execution stops only in goal states, which provides an incentive to achieve the goal. We then develop efficient methods that transform planning tasks with exponential utility functions to planning tasks that existing planners from artificial intelligence can solve. The transformations are such that an optimal plan for the transformed planning task is also optimal for the original one, and a good ("satisficing") plan for the transformed planning task is also good ("satisficing") for the original one. Thus, the transformed planning task can be solved with both optimal and suboptimal ("satisficing") planners. We first describe the additive planning-task transformation and then

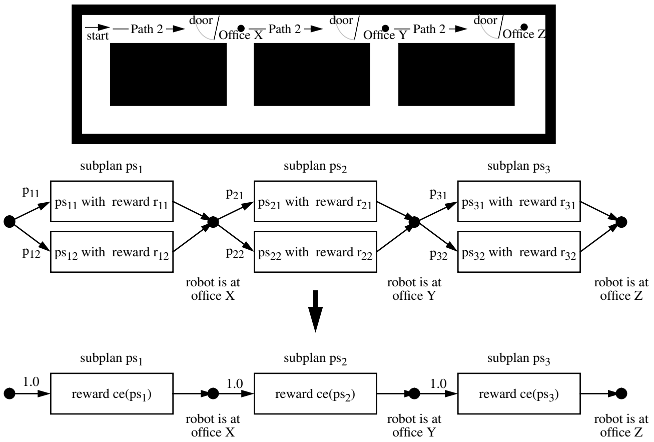

Figure 3: Additive Planning-Task Transformation

the multiplicative planning-task transformation. The additive planning-task transformation applies to special cases of planning tasks and transforms them to deterministic planning tasks. The multiplicative planning-task transformation applies to arbitrary probabilistic planning tasks and transforms them to other probabilistic planning tasks. Thus, the multiplicative planning-task transformation has the advantage that it applies to a larger class of planning tasks. It has the disadvantage that it can amplify the errors of planners that can find only near-optimal plans for the transformed planning tasks, whereas the additive planning-task transformation leaves the errors unchanged. (We do not prove this particular property in this paper. It is also possible to apply the planning-task transformations to concave exponential utility functions, but this is more complicated than applying them to convex exponential utility functions and thus we do not discuss it either.)

3.1 The Additive Transformation

The additive planning-task transformation applies to special cases of planning tasks. These planning tasks consist of a goal and a collection of atomic subplans ("actions") that can be used as building blocks to construct plans. Each subplan starts in only one state (which is no restriction) and is "deterministic" in the sense that its execution always ends in the same state although it can result in different subchronicles and thus rewards. The start and end states of the subplan can be different, and the start and end states of different subplans can be different as well. These requirements are similar to assumptions sometimes made in the literature [Wellman *et al.*, 1995]. The *additive planning-task transformation* converts the planning task by modifying all of its subplans (everything else remains the same): If a subplan can be executed in state s and its execution leads with probability p_i and reward r_i to state s' (for all i), then it is replaced with a deterministic subplan. This subplan can be executed in state s and its execution leads with probability one and reward $u^{-1}(\sum_i [p_i u(r_i)])$ to state s'. This reward is the certainty equivalent of the original subplan for the convex exponential utility function. We use the planning task from Figure 3 (top) to illustrate the additive planning-task transformation. The planning task consists of three parts. The robot has to first visit office X to pick up a form, then obtain a signature in office Y, and finally deliver the signed form to office Z, at which point plan execution stops. We assume that the probabilities of any two different doors being open are independent (a reasonable assumption). One subplan of getting from office X to office Y (marked "Path 2" in the figure) is to move on a shortest path to office Y and, if the door is closed, return to office X and take the long path to office Y. This subplan can be executed when the robot is at office X and always moves the robot to office Y, although its execution can result in different travel times. If the robot reaches office Y in 120.00 seconds with probability 0.50 and in 576.00 seconds with probability 0.50, then the subplan is replaced with a deterministic subplan that can be executed when the robot is at office X and whose execution moves the robot with probability one and reward $u^{-1}(0.50\,u(-120.00) + 0.50\,u(-576.00)) = \log_\gamma(0.50\,\gamma^{-120.00} + 0.50\,\gamma^{-576.00})$ to office Y. The other subplans are transformed similarly. The transformation is such that a plan with maximal total reward for the transformed planning task also maximizes the average total utility for the original planning task. We now explain why the additive planning-task transformation works.

Among all plans with maximal average total utility for the original planning task, there is always a *sequential plan*, that is, an unconditional sequence of subplans. This is so because the execution of every subplan always ends in only one state and exponential utility functions satisfy the Markov property, which implies that, for acting optimally in the future, it is unimportant how the current state has been reached or how much reward has already been accumulated. Consequently, planning can be restricted to sequential plans without giving up optimality. Now consider the certainty equivalent of any sequential plan for the original planning task. As an example, we use the plan in Figure 3 (center). It solves the planning task from Figure 3 by trying to take the route through the door (Path 2) for all three parts of the planning task. The first subplan ps_1 corresponds to reaching office X. It has two possible subchronicles ps_{11} and ps_{12}. Subchronicle ps_{11} denotes the case where the door is open, whereas subchronicle ps_{12} corresponds to the door being closed. Subchronicle ps_{11} has reward r_{11} and occurs with probability p_{11}, subchronicle ps_{12} has reward r_{12} and occurs with probability p_{12} (where $p_{11} + p_{12} = 1$), and similarly for the other subplans and subchronicles. Then, the certainty equivalent of the overall plan $ps_1 \cdot ps_2 \cdot ps_3$ (the concatenation of the three subplans) for the original planning task is

$$
\begin{aligned}
& ce(ps_1 \cdot ps_2 \cdot ps_3) \\
&\quad = \quad u^{-1}(eu(ps_1 \cdot ps_2 \cdot ps_3)) \\
&\quad = \quad u^{-1}(\sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{2}[p_{1i}p_{2j}p_{3k}u(r_{1i} + r_{2j} + r_{3k})]) \\
&\quad = \quad \log_\gamma \sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{2}[p_{1i}p_{2j}p_{3k}(\gamma^{r_{1i}+r_{2j}+r_{3k}})] \\
&\quad = \quad \log_\gamma \sum_{i=1}^{2}[p_{1i}(\gamma^{r_{1i}})] + \log_\gamma \sum_{j=1}^{2}[p_{2j}(\gamma^{r_{2j}})] \qquad (1) \\
&\qquad\qquad + \log_\gamma \sum_{k=1}^{2}[p_{3k}(\gamma^{r_{3k}})] \\
&\quad = \quad u^{-1}(eu(ps_1)) + u^{-1}(eu(ps_2)) + u^{-1}(eu(ps_3)) \\
&\quad = \quad ce(ps_1) + ce(ps_2) + ce(ps_3). \qquad (2)
\end{aligned}
$$

Instead of calculating its certainty equivalent directly, we can first transform the sequential plan. Its structure remains unchanged, but all of its subplans are transformed (as described above). The total reward of the transformed plan is the same as the certainty equivalent of the original plan. Figure 3 (bottom), for example, shows the transformation of the plan in Figure 3 (center). The total reward of the transformed plan is $ce(ps_1) + ce(ps_2) + ce(ps_3)$, which is also the certainty equivalent of the original plan according to Formula 2.

To determine plans with maximal average total utility for the original planning task, one can transform all subplans of the planning task. Then, a plan with maximal total reward for the transformed planning task also maximizes the certainty equivalent and thus the average total utility among all sequential plans for the original planning task, and (as we have argued) it is impossible for a nonsequential plan to have a larger average total utility. To summarize, the original planning task can be solved by applying the additive planning-task transformation and then solving the transformed planning task with any planner with preference model C2 and thus also with any planner with preference model C4.

In the following, we give two examples, both delivery tasks, that can be solved with the additive planning-task transformation. The first example is the planning task from Figure 3. The planning task specifies the subgoals, the ordering in which they have to be achieved, and the subplans that can be used to achieve them. This planning task satisfies the assumptions of the additive planning-task transformation because all subplans are able to move the robot to its destination, although the robot can take different paths and thus incur different travel times. The additive planning-task transformation applies as follows:

$$\begin{aligned}
&\max_{x,y,z} ce(ps_1(x)\cdot ps_2(y)\cdot ps_3(z))\\
&\overset{\text{Formula 2}}{=} \quad \max_{x,y,z}[ce(ps_1(x)) + ce(ps_2(y)) + ce(ps_3(z))]\\
&= \quad \max_{x} ce(ps_1(x)) + \max_{y} ce(ps_2(y)) + \max_{z} ce(ps_3(z)).
\end{aligned}$$

Thus, to find a plan with maximal average total utility, a planner can determine separate subplans for the three parts of the planning task, each of which maximizes the average utility of its part. The three subplans combined then form a plan with maximal average total utility for the planning task. In this case, the parts of the planning task can be solved completely independently. This is no longer possible in the next example, that is similar to the first one: This time, the robot starts at the secretary's office with the task of collecting ten signatures on a form and returning it to the secretary. We assume that all ten people are in their offices. This planning task is essentially one of task sequencing: we have to determine the order in which to visit the ten offices. The planning task specifies the subgoals and the subplans that can be used to achieve them, but not the ordering in which the subplans have to be achieved. The additive planning-task transformation converts the planning task to a traveling salesman problem on a directed graph with deterministic edge rewards. The reward of an edge is the certainty equivalent of the navigation plan with maximal average utility between the corresponding two locations. The traveling salesman problem can then be solved with traveling salesman problem (TSP) methods or traditional search and planning methods from artificial intelligence.

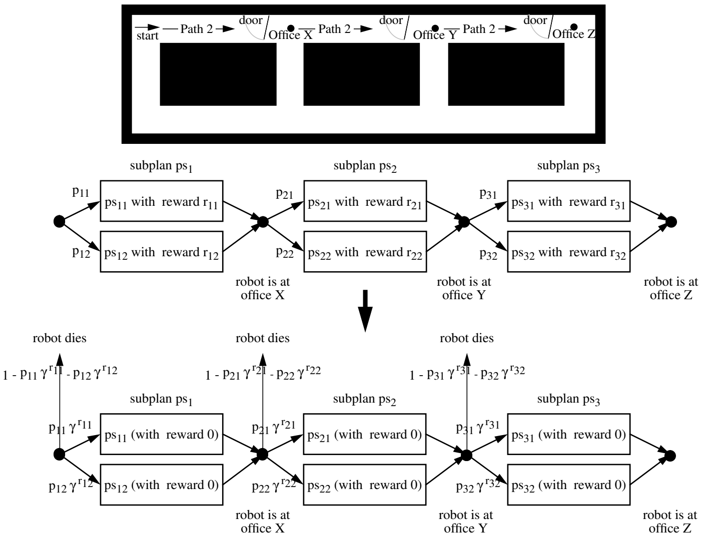

Figure 4: Multiplicative Planning-Task Transformation

While the additive planning-task transformation is useful, it cannot be used to solve all planning tasks because it is often not the case that each subplan is atomic and always ends in the same state. As an example, consider again a robot that has to collect ten signatures. The subplans move the robot to a specified office. They are not atomic in practice because their execution can be interrupted. For example, while the robot moves to some office, it might have to take a detour because a door is closed. If this detour leads past another office, it can be advantageous for the robot to go first to that office in order to obtain another signature on the way to its original destination. This suggests considering "move to the door" as a subplan because this would allow the robot to re-plan and change its destination when it recognizes that the door is closed. However, this subplan can end in two different states: the robot being at the door and the door being either open or closed. Thus, it does not satisfy the assumption of the additive planning-task transformation and we have to consider ways of finding *conditional* plans with maximal average total utility for convex exponential utility functions.

3.2 The Multiplicative Transformation

The multiplicative planning-task transformation is more general than the additive planning-task transformation. It applies to planning tasks that can be solved optimally with conditional plans. The planning tasks again consist of a goal and a collection of atomic subplans ("actions") that can be used as building blocks to construct plans, but it is no longer required that the execution of each subplan end in only one state. The *multiplicative planning-task transformation* converts the planning task by modifying all of its subplans (everything else remains the same): If a subplan can be executed in state s and its execution leads with probability p_i and reward r_i to state s_i (for all i), then it is replaced with a subplan that can be executed in state s and whose execution leads with probability $p_i\gamma^{r_i}$ to state s_i (for all i) and with probability $1 - \sum_i[p_i\gamma^{r_i}]$ to a new nongoal state ("death") in which plan execution stops (alternatively, we can let the agent cycle forever without receiving any future rewards). The rewards do not matter. As an example consider again the planning task from Figure 4 (top). One of the subplans for moving from office X to office Y is to move on a shortest path to office Y and, if the door is closed, return to office X and take the long path to office Y. If the robot reaches office Y in 120.00 seconds with probability 0.50 and in 576.00 seconds with probability 0.50, then the subplan is replaced with a subplan that can be executed when the robot is at office X and whose execution moves the robot with probability $0.5\gamma^{-120.00} + 0.5\gamma^{-576.00}$ to office Y and with probability $1 - 0.5\gamma^{-120.00} - 0.5\gamma^{-576.00}$ to the new nongoal state "death." The other subplans are transformed similarly. The transformation is such that a conditional plan with maximal probability of goal achievement for the transformed planning task also maximizes the average total utility for the original planning task. We now explain why the multiplicative planning-task transformation works.

We first consider sequential plans as a special case and then conditional plans in general. As an example, we use again the sequential plan $ps_1 \cdot ps_2 \cdot ps_3$ from Figure 4 (center). It solves the planning task from Figure 3 by trying to take the route through the door (Path 2) for all three parts of the planning task. Its average total utility for the original planning task is

$$
\begin{aligned}
& eu(ps_1 \cdot ps_2 \cdot ps_3) \\
= & \sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{2}[p_{1i}p_{2j}p_{3k}u(r_{1i}+r_{2j}+r_{3k})] \\
= & \sum_{i=1}^{2}\sum_{j=1}^{2}\sum_{k=1}^{2}[p_{1i}p_{2j}p_{3k}\gamma^{r_{1i}+r_{2j}+r_{3k}}] \\
= & \sum_{i=1}^{2}[p_{1i}\gamma^{r_{1i}}] \times \sum_{j=1}^{2}[p_{2j}\gamma^{r_{2j}}] \times \sum_{k=1}^{2}[p_{3k}\gamma^{r_{3k}}] \\
= & \sum_{i=1}^{2}\overline{p}_{1i} \times \sum_{j=1}^{2}\overline{p}_{2j} \times \sum_{k=1}^{2}\overline{p}_{3k}, \qquad (3)
\end{aligned}
$$

where the parameters $\overline{p}_{mn}$ are new values with $\overline{p}_{mn} := p_{mn}\gamma^{r_{mn}}$. These values satisfy $0 \leq \overline{p}_{mn} \leq p_{mn}$ according to our assumption that the exponential utility function is convex ($\gamma > 1$) and all rewards are negative ($r_{mn} < 0$). Thus, the parameters $\overline{p}_{mn}$ can be interpreted as probabilities. Instead of calculating the average total utility of the sequential plan directly, we can first transform the plan. Its structure remains unchanged, but all of its subplans are transformed (as described above). The average total utility of the original plan is the same as the probability of not dying while executing the transformed plan, which is the product of the probabilities of not dying while executing its subplans (due to probabilistic independence). Figure 4 (bottom), for example, shows the transformation of the plan in Figure 4 (center). The probability of not dying during the execution of the transformed plan is $\sum_{i=1}^{2}[p_{1i}\gamma^{r_{1i}}] \times \sum_{j=1}^{2}[p_{2j}\gamma^{r_{2j}}] \times \sum_{k=1}^{2}[p_{3k}\gamma^{r_{3k}}]$, which is also the average total utility of the original plan according to Formula 3.

Now consider an arbitrary conditional plan for the original planning task. We can use the multiplicative planning-task transformation on the conditional plan and it remains true that the average total utility of the original conditional plan is the