A. Blum and M. Furst, "Fast Planning Through Planning Graph Analysis", Artificial Intelligence, 90:281--300 (1997).

This paper explains a Planning Graph, and a new planner Graphplan guarantees that it returns a shortest-possible partial-order plan or states that no valid plan exists. Another advantage of Graphplan is that it is not sensitive to the order of the goals in a planning task unlike traditional approaches. In experiments where the performance of the algorithm was compared with two popular planners, Prodigy and UCPOP, to solve planning problems, it achieved the best running time and also was able to find the shortest plan when the other two planners failed to find a solution.

One feature of Planning Graph is to use mutual exclusions among nodes. Graphplan tries to find mutual exclusion relationships using a few simple rules. Those rules do not guarantee to find all mutual exclusion relationships, but find many of them. Interference and Competing Needs are the two ways in which actions a and b are marked to be exclusive of each other.

Graphplan has a few limitations however. First, it applies only to STRIP-like domains. The paper used an example where one of the actions allows the planner to dig a hole of an arbitrary integral depth. In this case we have infinite number of objects that can be created, and Graphplan cannot find a solution. Second, it requires either one of the following conditions; i) the mutual exclusion relations capture important constraint of the problem; ii) the ability to perform parallel actions reduces the depth of the graph. In a problem where neither of the conditions is met, it may perform poorly. Third, Graphplan can make problems more difficult for itself by guaranteeing to find the shortest plan. In some problems we may not need the shortest plan as long as the solution is almost as good as the shortest plan, and we may prefer faster computation.