OPTIMIZING BACKTRACK SEARCH FOR ALL SOLUTIONS TO CONJUNCTIVE PROBLEMS

K. S. Natarojon

IBM Thomas J. Watson Research Center
P.O. 704, Yorktown Hoights, NY 10598

Abstract

We consider the problem of minimizing depth-first search effort for the generation of all solutions to a problem stated as a conjunction of subproblems. For a sequence of subproblems that share no variables, the effort is minimized by ordering the subproblems in decreasing ratio of *NC/(N - 1)*, where *N* and C are the number of solutions and the search effort of obtaining a solution to the subproblem. If a conjunctive problem has an arbitrary number of subproblems sharing variables among them, we assume that in the solution sets of the subproblems, each argument variable is bound to elements of its domain with equal frequency. Under this uniform distribution assumption, we derive a set of necessary conditions that must be satisfied by an optimal depth-first sequence. If the distribution assumption does not hold for a conjunctive problem, then the search effort can be optimized only if the sequencing of subproblems is suitably interleaved with the actual enumeration of solutions to the problem.

1. Introduction

In this paper we consider the problem of optimization of search effort for the generation of all solutions to a problem that is stated as a conjunction of subproblems. Conjunctive problems arise frequently in many applications of Artificial Intelligence as well as in database applications of logic. Depth-first search with backtracking is a commonly used problem-solving search technique used in such applications. it is the default strategy for the control of inference in sequential Prolog. Our motivation here is to develop search ordering conditions for the optimal control of depth-first search required in such problem-solving systems. Knowing what constitutes optimal ordering conditions or a characterization of the conditions that can be used systematically to improve a given ordering is critical for optimizing the efficiency of problem-solving systems.

A conjunctive problem has the generic form:

$$G(S) = P_1(S_1) \ \& \ P_2(S_2) \ \& \ \dots \ \& \ P_m(S_m) \qquad (1)$$

where, $G(S)$ is the problem to be solved and $S = \{X_1, X_2, \dots, X_n\}$ is a set of n argument variables. For $1 \le i \le m$, $P_i(S_i)$ is the i^{th} subproblem with arguments S_i, where $S_i \subseteq S$. A solution to problem G is a binding of constants to each of the arguments of $P_1, P_2, \dots, P_m$ such that a) each subproblem P_i is solved, and b) each argument that is common to two or more subproblems must be consistently bound to a constant value. Two solutions are distinct if they have different bindings, i.e., in one solution there is at least one argument whose binding is different in the other solution.

It has been recognized [SMI85a] that a significant weakness of backtracking is that the computational effort in searching for all feasible solutions to a conjunctive problem is quite sensitive to the sequence in which the subproblems are considered. In the next section, we review related work on controlling the search required to generate all solutions to conjunctive problems. In Section 3, we present our results on how to order subproblems such that the search effort for generating all solutions is minimized. The results are summarized as follows. We show the conditions for optimal ordering of subproblems of a conjunctive problem that has: a) many independent subproblems, or b) exactly two dependent subproblems. For conjunctive problems with an arbitrary number of dependent subproblems, we derive necessary conditions for optimal ordering. The conditions can be used for systematic transformation of a given ordering of subproblems into a locally optimal sequence of subproblems. In Section 4 we present our conclusions.

2. Related Work

The problem we consider here, namely, the efficient generation of all solutions to a conjunctive problem with backtracking as the basic problem-solving strategy, has been addressed in the past by a number of researchers. The work of Warren [WAR81], Smith [SMI85a] and Smith and Genesereth [SMI85b] are especially pertinent to us.

Warren [WAR81] considered the efficient evaluation of database queries, which may be regarded as a class of conjunctive problems, expressed in the Prolog language. Warren developed a heuristic approach for improving the efficiency by a suitable reordering of the subproblems of a conjunctive problem. The approach, demonstrated experimentally in a system for natural language question answering, relies on the availability of statistical information such as the number of solutions to each subproblem, the number of distinct values for each argument in a solution, etc. Two important assumptions are made in [WAR81). The first assumption made is that the uniform distribution of attribute values holds. The uniform distribution assumption holds if in the solution set of a problem, each argument variable of the problem is bound to elements of its domain with equal frequency.

The second assumption made implicitly in [WAR81] is the uniform cost assumption. The assumption is that the cost of obtaining a solution to a subproblem is the same across all subproblems, i.e., each subproblem is computationally just as hard as any other subproblem in the conjunctive problem. We note that there are many applications for which the uniform cost assumption does not hold. To illustrate this, consider a conjunctive problem consisting of three subproblems, say, P_1, P_2 and P_3. Suppose the solutions to Subproblem P_1 are explicitly available and stored in the main memory of a computer; solutions to Subproblem P_2 are also explicitly available but stored in a slower peripheral device, such as a disk; and, solutions to P_3 are not explicitly available but a problem-solving search procedure can determine the solutions after possibly applying certain inference procedures. In such a situation, the cost (or equivalently, the time needed) for obtaining a solution is very unlikely to be the same across different subproblems. For the purpose of efficient solution of conjunctive problems under such more general conditions, we relax the uniform cost assumption and allow the cost per solution depend on the subproblem being solved. We still assume, however, that for any given subproblem the cost of obtaining a solution is constant, i.e., it does not depend on the specific solution obtained for the subproblem.

Relevant analytical work on optimal static ordering of the subproblems of a conjunctive problem has recently been reported by Smith and Genesereth [SMI85a] [SMI85b] Static ordering means the planning, i e , the decision on how to sequence the subproblems, is made before the actual backtrack enumeration effort is undertaken Once an ordering of the subproblems is chosen it is left unchanged during the entire enumeration phase of problem-solving. Smith and Genesereth have addressed how a problem-solving system can minimize the effort in generating all solutions by using information provided to the system regarding average number of solutions to each subproblem The uniform distribution assumption regarding attribute values and the uniform cost assumption have also been made in [SM185b] The work cited above, based on assumptions of uniform cost and uniform distribution, provides a good starting point for us

We note that the total search effort in enumerating all the solutions to a conjunctive problem typically depends on both: a) how the values of each argument variable are distributed in the solution set of each subproblem, and b) the cost associated with the solution of each subproblem In this paper, we consider the impact of relaxing the assumptions of uniform cost and uniform distribution, on optimizing the search ordering.

3. Results

Consider the problem $G(S)$ stated in (1) Suppose the solutions to G are enumerated by solving the subproblems in the sequence shown in (1). Let N_i and C_i be respecnively the number of solutions and the cost of obtaining a solution to Subproblem P_i, given that subproblems P_1 through P_{i-1} have been solved. Assume the root of the search tree is at Level 0. There are N_1 solutions to P_1 and each solution is represented by a node at Level 1 Corresponding to each solution of P_1, there arc N_2 solutions of P_2 and hence, N_1N_2 nodes at Level 2 In general, at Level i of the search tree, there are $N_1N_2 \ldots N_i$ nodes. Thus, the total number of nodes in the backtrack search tree corresponding to the sequence expressed in (1) is:

$$1 + N_1(1 + N_2(1 + \cdots N_{m-1}(1 + N_m) \ldots)) \qquad (2)$$

Since there are N_1 nodes at Level 1, each requiring a search effort of C_1, the total search cost at Level 1 is N_1C_1. In general, the total search effort at Level i is $N_1N_2 \ldots N_iC_i$, the product of the number of nodes at Level i and the search effort per node at that level. Thus, the total cost of the backtrack search algorithm corresponding to the sequence expressed in (1) is :

$$N_1(C_1 + (N_2(C_2 + \cdots N_{m-1}(C_{m-1} + N_mC_m) \ldots))) \qquad (3)$$

3.1 Independent Subproblems

A Subproblem $P_i(S_i)$ is defined to be *independent* of Subproblem $P_j(S_j)$ (for $i \neq j$) if S_i has no variable in common with S_j. Two subproblems that share at least one common variable are defined to be *directly dependent.* Two Subproblems, P_i, P_j $(i \neq j)$, are considered *indirectly dependent* if there exists a Subproblem P_k $(k \neq i,j)$ such that P_i is directly dependent on P_k and P_k is dependent, either directly or indirectly, on P_j. For a sequence of independent subproblems, the following result shows the conditions that must be satisfied by an optimal ordering of subproblems. The result holds regardless of the distributions of argument variables The proof can be found in [NAT86]

Result Let a conjunctive problem, expressed as in (1), have Subproblems $P_1(S_1)$ through $P_m(S_m)$ that are pairwisc independent, i.e., no subproblem shares a variable with any other subproblem. Then the ordering that minimizes (3), the cost of generating all solutions to the problem, is to sequence the subproblems such that:

$$\left(\frac{N_i}{N_i - 1}\right)C_i > \left(\frac{N_j}{N_j - 1}\right)C_j$$

for $i \neq j$, $1 \leq i,j \leq m$

Special Case (i). If each subproblem has the same number of solutions (i.e., $N_1 = N_2 = \cdots = N_m$),then the subproblems must be ordered in decreasing order of their costs, i.e., Subproblem P_i should be solved before Subproblem P_j whenever $C_i > C_j$

Special Case (ii). If the cost of obtaining a solution is identical for the subproblems (i.e., $C_1 = C_2 = \cdots = C_m$), then the subproblems must be ordered in ascending order of their number of solutions, i.e., Subproblem P_i should be solved before Subproblem P_j whenever $N_i < N_j$.

It is significant to note that the above result can often be used *even* if, in practice, a given conjunctive problem expressed in the form of (1) may have the property that every pair of subproblems is dependent, cither directly or indirectly, on each other. A problem-solver can use the knowledge about optimal ordering conditions for independent subproblems even if the given conjunctive problem contains only dependent pairs of subproblems.

3.2 Dependent Subproblems

Typically, the number of solutions to $P_i(S_i)$ depends on all the subproblems that were solved prior to P_i. The objective of minimizing the search cost is equivalent to choosing the best permutation of the set of subproblems such that the cost expressed in (3) is minimized. A conjunctive problem is considered feasible if there exists at least one solution to the problem.

3.2.1 Two Subproblems in a Conjunctive Problem

We first consider conjunctive problems expressed as conjunctions of exactly two subproblems. In this section, the uniform distribution assumption and the uniform cost assumption are both relaxed. Consider a problem stated as a conjunction of two subproblems: $P_1(S_1)$ & $P_2(S_2)$, where the cost of obtaining a solution to Subproblems P_1 and P_2 are C_1 and C_2 respectively. Without loss of generality, assume that S_1 and S_2 have exactly one variable X common to them. Let $D(X) = \{v_1, v_2, \ldots, v_K\}$ be the domain of X with cardinality K. X can be bound to one of the K elements. Among the solutions to Subproblems $P_1(S_1)$ and $P_2(S_2)$, let X be bound to v_j with frequencies N_{1_j} and N_{2_j} respectively. Let R_1 represent the sequence Subproblem P_1 followed by Subproblem P_2 and R_2 represent the sequence Subproblem P_2 followed by Subproblem P_1. Let $Cost(R_1)$ and $Cost(R_2)$ respectively denote the cost of searching for all solutions by using R_1 and R_2. The cost of search for the two different sequences is computed as follows. Let $T_{1,2}$ be the number of solutions to the conjunctive problem.

$$Cost(R_1) = \sum_{j=1}^{K} N_{1_j} C_1 + \sum_{j=1}^{K} N_{1_j} N_{2_j} C_2 = N_1 C_1 + T_{1,2} C_2$$

$$Cost(R_2) = \sum_{j=1}^{K} N_{2_j} C_2 + \sum_{j=1}^{K} N_{2_j} N_{1_j} C_1 = N_2 C_2 + T_{1,2} C_1$$

Thus, we have the following result.

$$Cost(R_1) < Cost(R_2) \Leftrightarrow \frac{C_1}{C_2} < \frac{N_2 - T_{1,2}}{N_1 - T_{1,2}} \qquad (4)$$

From (4) we obtain the following results for special cases. Case (i): If the search cost per solution of each subproblem is the same (i.e., $C_1 = C_2$), then the subproblem with fewer solutions should be solved first. Case (ii): If the two subproblems have an equal number of solutions (i.e., $N_1 = N_2$) and the conjunctive problem has at least N, solutions, then the subproblem that is more costly should be solved first; otherwise., the less costly problem should be solved first.

3.2.2 At Least Three Subproblems in a Conjunctive Problem

We next consider conjunctive problems that are expressed as a conjunction of three or more subproblems. We relax the uniform cost assumption but assume the uniform distribution assumption holds for the individual subproblems. If a subproblem has *t* solutions, and *X* is one of its argument variables with domain $D(X) = \{v_1, v_2, \ldots, v_K\}$, our assumption is that for any specific value of *X,* there will be exactly *t/K* solutions to the subproblem.

Suppose a conjunctive problem expressed as in (1) is solved by ordering the subproblems in two different sequences, R_1 and R_2 as follows:

$$R_1\!: \; P_1(S_1) \;\&\; \ldots \;\&\; P_i(S_i) \;\&\; P_{i+1}(S_{i+1}) \;\&\; \ldots \;\&\; P_m(S_m)$$

$$R_2\!: \; P_1(S_1) \;\&\; \ldots \;\&\; P_{i+1}(S_{i+1}) \;\&\; P_i(S_i) \;\&\; \ldots \;\&\; P_m(S_m)$$

The subproblems solved at each level of the search tree traversed by a backtracking algorithm is shown in Fig. 1. Note that each level of the search tree corresponds to exactly one of the *m* distinct subproblems of the conjunctive problem. The two search sequences, R_1 and R_2 differ in only the relative ordering of Sub-

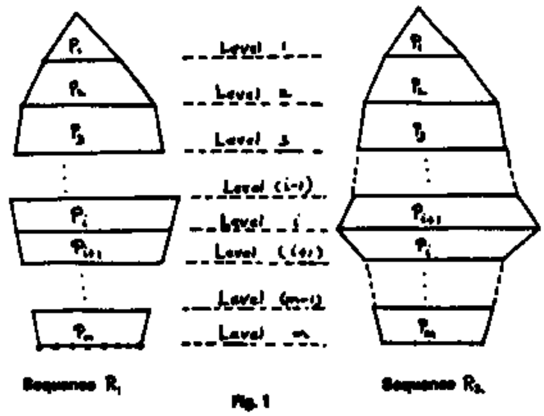

Fig. 1

problems P_t and P,. Consequently they spend exactly the same computational effort at all levels except at Level / and Level (i + 1) of the search tree. For a given ordering of the subproblems, let *Cost(Levelj)* be the search cost incurred at Level *j* of the search tree corresponding to the ordering. Note that *Cost(Leyelj)* is the product of the total number of attempts to find a solution to the subproblem considered at Level *j* and the average cost per solution attempt of the subproblem. We have,

$$Cost\,(R_1) = \sum_{j=1}^{m} \left[Cost\,(Level_j) \;\; using \;\; R_1 \right]$$

$$Cost\,(R_2) = \sum_{j=1}^{m} \left[Cost\,(Level_j) \;\; using \;\; R_2 \right]$$

Let *Q* denote the total number of partial solutions to the the first (/ - 1) subproblems of either sequence. We have to distinguish two possible cases that may arise.

Cased): The set of variables in *S,* that are bound when P_t is solved is disjoint from the set of variables in S_{1+1} that are bound when P_{1+1} is solved.

Cased!): The set of variables in *S,* that are bound when *P,* is solved contains at least one variable that also occurs in S_{1+1}.

Let *a,* denote the number of solutions to *P,* given that the initial sequence of subproblems, $P_1, P_2, \ldots, P_{i-1}$, has been solved. Let a_{1+1} denote the number of solutions to P^{1+1} given that the initial sequence of subproblems,P_1 ....,P_1 has been solved.

Case(i): The difference in search costs of sequences R^1 and R_2, $Cost(R_1)$ - $Cost(R_2)$, equals:

$$(Qa_iC_i + Qa\rho_{i+1}C_{i+1}) - (Qa_{i+1}C_{i+1} + Qa\rho_{i+1}C_i)$$

Thus, Sequence R_1 is better than Sequence R_2 (i.e., $Cost(R_1) < Cost(R_2)$), if and only if,

$$C_i(a_i - a\rho_{i+1}) < C_{i+1}(a_{i+1} - a\rho_{i+1}) \qquad (5)$$

From the condition expressed in (5) we obtain the following resuits for ininimizing the average search cost.

a) If the search cost per solution of Subproblems P_i and P_{i+1} is the same (i.e., $C_i = C_{i+1}$), then the subproblem with fewer solutions should be solved first.

b) If the two subproblems have equal number of solutions (i.e., $a_i = a_{i+1}$) and the conjunctive problem is feasible, then the subproblem that is more costly should be solved before the less costly problem.

If neither the number of solutions nor the cost *per* solution of Subproblems P_i and P_{i+1} are equal, then Subproblem P_i should be considered before P_{i+1} whenever:

c) P_i is more costly than P_{i+1} and the condition $\frac{C_i}{a_i} > \frac{C_{i+1}}{a_{i+1}}$ holds.

d) P_i is less costly than P_{i+1} and the condition $a_iC_i < a_{i+1}C_{i+1}$ holds.

In the general case, the decision about whether Subproblem P_i should be solved before P_{i+1} is resolved by evaluating the condition expressed in (5).

Case(iii): The size of the domain of a variable is the cardinality of the set of distinct values that can be bound to the variable. Let *s* represent the product of domain sizes of each variable that occurs in both S_i and S_{i+1} and has not already been bound during solution of the sequence $P_1, P_2, \ldots, P_{i-1}$. The difference, $Cost(R_1) - Cost(R_2)$, in search costs of sequences R_1 and R_2 that differ in only the work done at Levels i and $(i+1)$ equals:

$$\left(Qa_iC_i + \frac{Qa\rho_{i+1}C_{i+1}}{s}\right) - \left(Qa_{i+1}C_{i+1} + \frac{Qa\rho_{i+1}C_i}{s}\right)$$

Thus, Sequence R_1 is better than R_2, if and only if,

$$C_i\left(a_i - \frac{a\rho_{i+1}}{s}\right) < C_{i+1}\left(a_{i+1} - \frac{a\rho_{i+1}}{s}\right) \qquad (6)$$

From the condition expressed in (6) we obtain the results analogous to (a), (b), (c), and (d) in case (i) above for minimizing the average search cost. In the general case, the decision about whether Subproblem P_i should be solved before P_{i+1} is resolved by evaluating the condition expressed in (6).

The conditions for optimal ordering of subproblems derived in this section (3.2.2) are based on the key assumption of uniform distribution of arguments of each subproblem. The uniformity of distribution means that the structure of the search tree has the following characteristic. At any level of the search tree, the number of sons of each node at that level is the same. Ordering of subproblems has an impact on the degree (i.e., number of sons) of nonterminal nodes at each level of the search tree. At each specific level of the tree, the degree is uniform. Search ordering that minimizes total search effort is one of the ml possible sequences in which *m* subproblems may be statically sequenced before the enumeration effort begins.

We briefly discuss what will be the impact of non-uniform distribution of arguments of subproblems. Suppose one is interested in generating all solutions to a conjunctive problem for which the uniform distribution assumption does *not* hold. Typically, search effort for such a problem can not be optimized by restricting to the class of static orderings of subproblems (see [NAT86] for details). For such problems, the minimization of search effort requires that search ordering be context-dependent, i.e., whenever a decision has to be made on what subproblem to solve next, it must take into account the existing binding environment at that instant. With this technique, the planning (i.e., sequencing of subproblems) is interleaved with the enumeration of solutions. This technique of *dynamic* search ordering or search rearrangement [BIT75] has been effectively used in the solution of combinatorial puzzles. The technique holds potential for efficient solution of conjunctive problems in general problem-solving applications that are characterized by non-uniform distributions.

4. Conclusions

In general, an optimal depth-first search sequence, that minimizes the total computational effort for solving conjunctive problems, depends on both: a) how the values of each argument variable are distributed in the solution sets of the subproblems, and b) the cost associated with the solution of each subproblem.

For a sequence of independent subproblems, we stated the optimal ordering conditions that can be used even if a given conjunctive problem has only dependent pairs of subproblems. For a sequence of three or more dependent subproblems, if the uniform distribution assumption holds and the uniform cost assumption is relaxed, then we showed search ordering conditions that can be used for systematic transformation of a given ordering of subproblems into a locally optimal sequence of subproblems. If assumptions regarding both the distributions and costs associated with subproblems are relaxed, then dynamic sequencing of subproblems is necessary to minimize the search effort. Issues related to the use of dynamic search ordering for improving search efficiency merit further study.

References

[WAR81] D. H. D. Warren, "Efficient Processing of Interactive Relational Database Queries Expressed in Logic", *Proc. of 7th VLDB Conference,* Cannes (September 1981), pp. 272-281.

[SM185a] D. E. Smith, "Controlling Inference", *Ph.D. Dissertation,* Stanford University, Stanford, CA 1985.

[SMI85b] *D.* E. Smith and M. R. Genesereth, "Ordering Conjunctive Queries", *Artificial Intelligence,* June 1985.

[NAT86] K. S. Natarajan, "On optimizing backtrack search for all solutions to conjunctive problems", *IBM Research Report 11982,* June 1986.

[BIT76] J. R. Bitner and E. M. Reingold, "Backtrack Programming Techniques", *Comm, ACM,* Nov. 1975, pp. 651-656.