Available online at www.sciencedirect.com
SCIENCE DIRECT®
Artificial Intelligence 155 (2004) 183–206
Artificial Intelligence
www.elsevier.com/locate/artint

Average-case analysis of best-first search in two representative directed acyclic graphs

Anup K. Sen [a], Amitava Bagchi [a], Weixiong Zhang [b,*]

[a] *Indian Institute of Management Calcutta, Joka, D. H. Road, Calcutta 700 104, India*
[b] *Department of Computer Science and Engineering, Washington University in Saint Louis, St. Louis, MO 63130, USA*

Received 5 May 2003

Abstract

Many problems that arise in the real world have search spaces that are graphs rather than trees. Understanding the properties of search algorithms and analyzing their performance have been major objectives of research in AI. But most published work on the analysis of search algorithms has been focused on tree search, and comparatively little has been reported on graph search. One of the major obstacles in analyzing average-case complexity of graph search is that no single graph can serve as a suitable representative of graph search problems. In this paper we propose one possible approach to analyzing graph search. We take two problem domains for which the search graphs are directed acyclic graphs of similar structure, and determine the average case performance of the best-first search algorithm A^* on these graphs. The first domain relates to one-machine job sequencing problems in which a set of jobs must be sequenced on a machine in such a way that a penalty function is minimized. The second domain concerns the Traveling Salesman Problem. Our mathematical formulation extends a technique that has been used previously for analyzing tree search. We demonstrate the existence of a gap in computational cost between two classes of problem instances. One class has exponential complexity and the other has polynomial complexity. For the job sequencing domain we provide supporting experimental evidence showing that problems exhibit a huge difference in computational cost under different conditions. For the Traveling Salesman Problem, our theoretical results reflect on the long-standing debate on the expected complexity of branch-and-bound algorithms for solving the problem, indicating that the complexity can be polynomial or exponential, depending on the accuracy of the heuristic function used.

Keywords: Graph search; Average-case complexity; A^*; Job sequencing; Traveling salesman

* Corresponding author.
E-mail addresses: sen@iimcal.ac.in (A.K. Sen), bagchi@iimcal.ac.in (A. Bagchi), zhang@cse.wustl.edu (W. Zhang).

0004-3702/$ – see front matter
doi:10.1016/j.artint.2004.01.001

1. Introduction and overview

In many real-world applications, graph search can be shown to be more efficient than tree search. Sometimes, tree search is simply not feasible. Consider sequence alignment, an important issue in computational biology that can be formulated as a shortest-path problem in a grid. It can be solved only with the help of graph search [5,12,17]. Problems that arise in single-machine minimum-penalty job sequencing can be solved with the help of either graph search or tree search, but in this case graph search outperforms tree search in terms of running time even when the evaluation function is non-order-preserving [26]. In addition, in the determination of a winner in a combinatorial auction [25], a solution can be obtained in principle by enumerating exhaustive partitions of items in a search space that has the form of a graph. These examples demonstrate that graph search has many possible applications. Moreover, graph search usually needs much less memory than best-first tree search, making larger problems solvable on current machines [27].

Although graph search plays an important role in understanding, characterizing and solving difficult combinatorial optimization problems, the average-case performance of graph search algorithms has hardly been analyzed at all. In sharp contrast, there is a large literature devoted to the analysis of the performance of tree search algorithms [2,6,7,11,16, 20–22,31]. One objective of this paper, which extends the work already reported in [28], is to try and redress the balance.

A major consideration that has hamstrung the research on the performance analysis of graph search algorithms is that no single graph can claim to be a representative graph for the search spaces that arise in real-world applications. Therefore, general results on the performance of graph search seem to be out of reach. One way out of this dilemma is to do a separate independent analysis of each individual problem, but this makes generalization difficult. Here we take a middle road. Our analysis is neither perfectly general nor confined to an individual case. We choose to study a representative model of a set of related problems, in the hope that this will not only shed more light on the individual problems, but perhaps also lead to a generalization that will help to achieve a deeper understanding of graph search.

Here we look at two classes of problems. The first class consists of one-machine job sequencing problems, and the second consists of the Traveling Salesman Problem (TSP). Job scheduling and job sequencing problems arise frequently in manufacturing and production systems as well as in information processing environments [23]. We consider a class of problems in which N given jobs must be sequenced on a machine in such a way that a specified function of job completion times is minimized. The function can have many different forms and might involve the minimization of measures such as the mean job lateness and/or earliness or the weighted sum of quadratic functions of completion times.

Our analytical model for the job sequencing problem has the form of a graph that defines a partial ordering on the subsets of a set of N elements (jobs) under the set inclusion property. N determines the *problem size*. The graph itself has 2^N nodes, and is a directed acyclic graph (DAG) with one root node, one goal node and multiple solution paths. The set of N elements forms the root node at level 0, and the empty set is the goal node at level N. To make the analysis feasible, it is assumed, following [22], that the normalized errors of heuristic estimates of nongoal nodes are independent, identically distributed (i.i.d.)

random variables. Using this abstract model, we analyze the complexity of the best-first graph search algorithm A*. The measure of complexity is the expected number of node expansions. We choose A* in preference to other search algorithms because it is optimal in terms of the number of node expansions among all algorithms that use the same heuristic information [8].

There are two main theoretical results. First, it is shown that under certain weak conditions the expected number of distinct nodes expanded by A* increases exponentially with the number N of jobs for large N. This is consistent with previous experimental results on the one-machine job sequencing problem [27]. Second, special cases of interest are identified in which the expected number of node expansions made by A* is polynomial in N for large N. The results indicate that the expected complexity of A* graph search on job sequencing problems has two distinct phases, one exponential and the other polynomial, demonstrating the existence of a huge gap similar to a phenomenon of phase transition. Experimental results support the theoretical analysis. We summarize the previous results for the exponential case and provide new test results on the polynomial case.

A similar, but not identical, search graph arises in a solution procedure for the Traveling Salesman Problem (TSP). In essence, all known algorithms for solving the TSP exactly are implicit enumeration methods based on the branch-and-bound procedure which progressively construct complete tours [9,10,18]. One difference between these algorithms relates to the branching rules that determine whether the search space is a tree or a graph [3]. Even though some of the best TSP algorithms use a tree search space, e.g., [30], it remains to be determined which search space, tree or graph, is more efficient. It is very likely that the search space to use would depend on the application at hand. In addition, a graph search space, originally proposed for the TSP in [22], has been used as a benchmark domain in some computational experiments [24]. Therefore, it is worthwhile to examine graph search spaces in the context of the TSP. It is interesting to note that the theoretical results derived for the job sequencing domain have their counterparts in the TSP domain, which means that A* may run in exponential or nonexponential or polynomial time in the number N of cities. We state and prove the corresponding theorems. These results may shed light on the long-standing debate on the expected complexity of specific branch-and-bound algorithms for solving the TSP in [4]. It has been argued that a branch-and-bound algorithm can find an optimal solution to the TSP in time polynomial in N under certain conditions [4], or in time exponential in N when such conditions are hard to satisfy [15,19]. When run on graphs, A* can be viewed as a special type of branch-and-bound algorithm that uses a best-first search strategy and exits as soon as a solution is found. Therefore the results in this paper seem to indicate that the expected complexity of a branch-and-bound algorithm for the TSP can be polynomial or nonexponential or exponential, depending both on the definition of the inter-city cost function and the accuracy of the heuristic function.

The paper is organized as follows. The application domains and their representative search graphs are introduced in Section 2. Basic concepts relating to graph search and the framework of our analysis are presented in Section 3. In Section 4 we examine the job sequencing problem in greater detail, derive the expected complexity of A*, and also present supporting experimental results. Section 5 is concerned with the TSP, and Section 6 concludes the paper. Proofs of theorems are supplied in Appendix A.

2. Application problem domains and representative graphs

2.1. One-machine job sequencing problems

We now describe the representative graphs that model the search spaces of the two application problem domains mentioned above. Consider the graph that defines the partial ordering of subsets of a set of N elements under the set inclusion property. Let S_N be the set $\{1, 2, \ldots, N\}$. The subsets of S_N under the partial ordering relation induced by set inclusion form a directed acyclic graph G_{js} of 2^N nodes. In this graph S_N is the root node at level 0, and the empty set $\{\}$ is the goal node at level N. The immediate successors of a node n are the various subsets of S_N obtained from n by removing one element. Thus a node corresponding to a subset of k elements has k immediate successors. G_{js} for $N = 4$ is shown in Fig. 1. Such a graph has the following characteristics:

(1) A node with i elements is at level $N - i$.
(2) There are $i!$ distinct paths to a node at level i.
(3) There are $C(N, i)$ nodes at level i.
(4) The total number of paths starting at the root and going up to level i is $C(N, i) \cdot (i!)$.

This graph arises in the solution of certain one-machine job sequencing problems, and exhibits top–down and left–right symmetry. Graphs with similar symmetries also arise in graph search formulations for the Traveling Salesman Problem [22] and in the winner determination problem in combinatorial auctions [25].

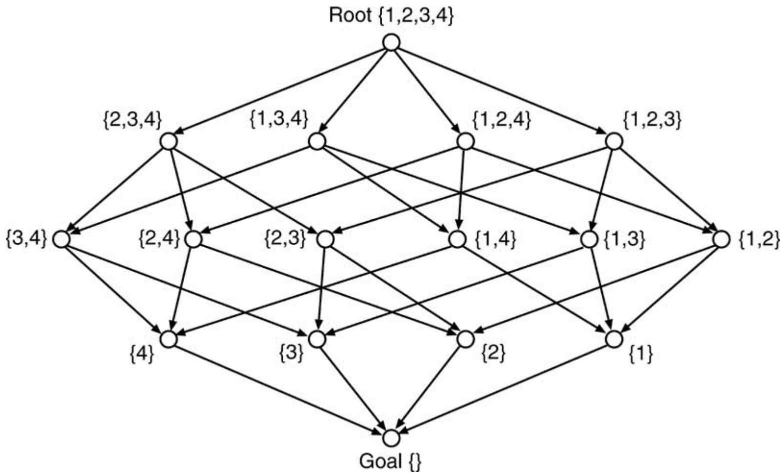

Fig. 1. Analytical model G_{js} for $N = 4$.

In the job sequencing problem, jobs J_i with processing times $A_i > 0$ for $1 \leqslant i \leqslant N$, are submitted to a one machine job shop at time 0. The jobs, which are assumed to have no setup times, are to be processed on the given machine one at a time. Let the processing of job J_i be completed at time T_i. Penalty functions H_i are supplied such that the penalty associated with completing job J_i at time T_i is $H_i(T_i) > 0$. H_i is nondecreasing in its argument, and in general nonlinear. The jobs must be sequenced on the machine in such a way that the total penalty $F = \sum_{i=1}^{N} H_i(T_i)$ is minimized. Nodes in the search graph G_{js} correspond to subsets of jobs that remain to be processed. The root node corresponds to the set of all N jobs, and the goal node to the empty set of jobs (i.e., to the completion of all jobs). Arc costs are assigned as follows. Suppose there is an arc from node n_1 to node n_2, and suppose job J_i is present in the subset of jobs associated with n_1 but absent from the subset of jobs at n_2. Then $c(n_1, n_2) = H_i(T_i)$. Here T_i is the time at which the processing of job J_i is completed; its value does not depend on the order in which jobs prior to job J_i are processed. Since setup times are ignored in this model, arc costs are order preserving [22]. Such graphs have been searched in the past using A* or TCBB [13]. Here we restrict ourselves to the A* algorithm because it is optimal in terms of node expansions [8].

2.2. Traveling Salesman Problem (TSP)

A graph-search solution procedure for the TSP gives rise to a similar search graph. Let the N cities be numbered $1, 2, \ldots, N$. Each node in the network is identified by a pair (the set of cities already visited, current city). The trip is assumed to begin from city 1, and the start node is $(\{\}, 1)$, where $\{\}$ represents the empty set. The goal node is $(\{1, 2, \ldots, N\}, 1)$. A* expands the start node and generates its $N - 1$ successor nodes $(\{1\}, 2), (\{1\}, 3), \ldots, (\{1\}, N)$. The cost of the arc from $(\{\}, 1)$ to a successor $(\{1\}, J)$, where $2 \leqslant J \leqslant N$, equals the cost of going from city 1 (the current city of the predecessor node) to city J (the current city of the successor node). The complete network G_{tsp} for $N = 4$ is shown in Fig. 2. For a specific cost matrix, a tour of minimum cost is found by running A* on the above search graph. The heuristic estimate at a nongoal node is computed using a minimum spanning tree heuristic [10]. The network G_{tsp} has the following characteristics:

(1) The start node is at level 0, while the goal node is at level N. A node at level i corresponds to a partial tour in which i cities have already been visited and the $(i+1)$th city has been reached.
(2) A node at level i has $(i - 1)$ incoming arcs for $1 < i \leqslant N$, and i incoming arcs for $i = 0, 1$. Similarly, a node at level i has $(N - i - 1)$ outgoing arcs for $0 \leqslant i < N - 1$, and $(N - i)$ outgoing arcs for $i = N - 1, N$.
(3) The total number of nodes at level i is

$$\begin{cases} i \cdot C(N-1, i) & \text{if } 1 \leqslant i < N, \\ 1 & \text{if } i = 0 \text{ or } i = N. \end{cases}$$

(4) The total number of paths entering a node at level i is

$$\begin{cases} (i-1)! & \text{if } 1 \leqslant i \leqslant N, \\ 0 & \text{if } i = 0. \end{cases}$$

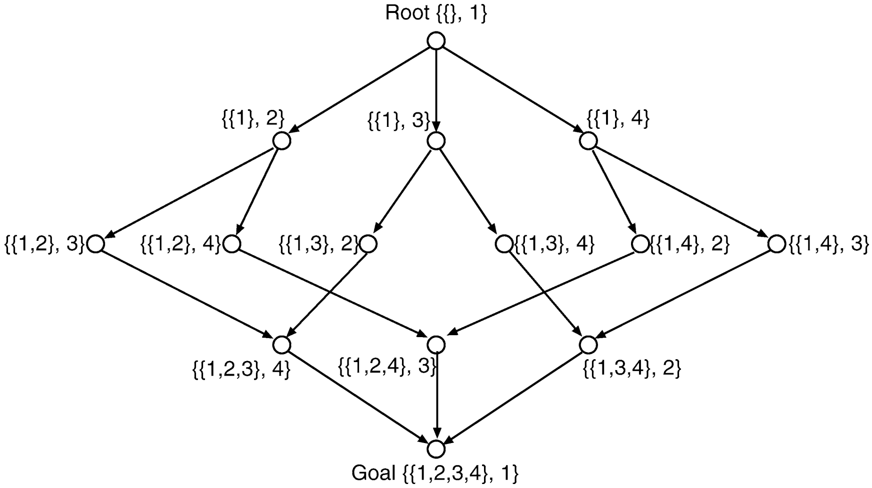

Fig. 2. Analytical model G_{tsp} for $N = 4$.

3. Basic concepts

A *search graph* (or *network*) G is a finite directed graph with nodes $n, n', n_1, n_2, \ldots$. The search always begins at the *start* (or *root*) node s, and ends at the *goal* node r. Each directed arc (n_1, n_2) in G has a finite *arc cost* $c(n_1, n_2) > 0$. A path is a sequence of directed arcs. A *solution path* is a path that begins at the start node s and ends at the goal node r. The cost $c(P)$ of a path P is the sum of the costs of the arcs that make up the path. The objective of a search algorithm like A* is to find a solution path of minimum cost in G. To find such a solution path, A* uses a nonnegative *heuristic estimate* $h(n)$ associated with each nongoal node n in G; $h(n)$ can be viewed as an estimate of $h^*(n)$, which is the cost of a path of least cost from n to the goal node.

Let $g^*(n)$ be the cost of a path of least cost from the start node to node n, and let $f^*(n) = g^*(n) + h^*(n)$. Then $f^*(n)$ is the cost of a solution path of least cost constrained to pass through node n. During an execution of A*, we use $g(n)$ to represent the cost of the path of least cost currently known from s to n. So $g(n)$ can be viewed as the current estimate of $g^*(n)$, and $f(n) = g(n) + h(n)$ as the current estimate of $f^*(n)$. As is customary, $f^*(r)$ denotes the cost of a minimum cost solution path in G.

Our networks are directed acyclic graphs. In such graphs, introducing more than one goal node does not add to the generality because there are multiple paths from the root node to the goal node in any case. When A* is run on such a network, a node may reenter OPEN from CLOSED; as a result, a node may get expanded more than once. Let Z_d and Z_t denote, respectively, the number of distinct nodes expanded by A* and the total number of node expansions made by A* when run on a given network G. $Z_t - Z_d$ is also sometimes referred to as the total number of reopenings of nodes. Our primary goal is to determine the expected values $E(Z_d)$ and $E(Z_t)$.

In order to assign a probability distribution on the heuristic estimates of nongoal nodes in G in a meaningful way, we adopt the notion of a *normalizing function* [22, p. 184].

A normalizing function Φ is a total function with the set of nonnegative real numbers as domain and the set of real numbers greater than or equal to 1 as range. It has the following properties:

(1) $\Phi(0) = 1$;
(2) $\Phi(x)$ is nondecreasing in x;
(3) $\Phi(x)$ is unbounded, i.e., the range of Φ has no finite upper bound.

We allow Φ to take one of three functional forms, viz., identity, less-than-linear and logarithmic:

$$\begin{cases} \Phi(x) = \max\{1, x\}: & \text{identity,} \\ \Phi(x) = \max\{1, x^{\delta}\}, \text{ for some } \delta,\ 0 < \delta < 1: & \text{less-than-linear,} \\ \Phi(x) = \max\{1, \ln(x)\}: & \text{logarithmic.} \end{cases}$$

The *normalized error* at a nongoal node n is $\chi(n) = (h(n) - h^*(n))/\Phi(h^*(n))$. We assume that for all nongoal nodes in G, the normalized errors are i.i.d. random variables. The normalizing function Φ determines the accuracy of the heuristic estimate function. When Φ is the identity function, the magnitude of the error $h(n) - h^*(n)$ is proportional to $h^*(n)$. The purpose of allowing other functions, such as logarithmic ones, for example, is to enable us to study the consequences of limiting the error $h(n) - h^*(n)$ to lower order values, implying greater accuracy of heuristic estimates. We use χ in place of $\chi(n)$, as the $\chi(n)$'s are identically distributed. Let $F_\chi(x) = p(\chi \leqslant x)$ be the cumulative probability distribution function of χ. Then $F_\chi(x)$ is nondecreasing in x. We allow $F_\chi(x)$ to have discontinuities. These must be left-discontinuities, since $F_\chi(x)$ by definition is right continuous. We do not assume any specific functional form for $F_\chi(x)$.

A heuristic function h is *admissible* if for every nongoal node n in the network G, $h(n) \leqslant h^*(n)$. Otherwise h is *inadmissible*. If $F_\chi(x) = 1$ for $x \geqslant 0$, then all nongoal nodes have admissible heuristic estimates with probability 1. Let $a_1 = \mathrm{lub}\{x \mid F_\chi(x) = 0\}$ and $a_2 = \mathrm{glb}\{x \mid F_\chi(x) = 1\}$. For $x < a_1$, $F_\chi(x)$ is identically 0, while for $x \geqslant a_2$, $F_\chi(x)$ is identically 1. The heuristic estimate function is admissible if $a_2 \leqslant 0$; it is *purely inadmissible* if $a_1 > 0$. When $\Phi(x)$ is the identity function, we must have $a_1 \geqslant -1$. Even when the heuristic estimate function is not admissible (i.e., when $a_2 > 0$), so long as $a_1 < 0$, a node n will be expanded by A* if $f(n) < f^*(r)$ and at least one of the immediate predecessors of n has been expanded. Note that by definition of a_1 and a_2, it must be the case that

(1) $h(n) \geqslant h^*(n) + a_1\Phi(h^*(n))$ with probability 1;
(2) $h(n) \leqslant h^*(n) + a_2\Phi(h^*(n))$ with probability 1.

A heuristic estimate function is *consistent* [22] if for every nongoal node n in the network G and every immediate successor n' of n, it is the case that $h^*(n) \leqslant c(n, n') + h^*(n')$. From now onwards, whenever a normalizing function is under discussion, we assume that the corresponding a_1 and a_2 are finite.

4. Analysis of one-machine minimum-penalty job sequencing problems

4.1. Exponential complexity

We begin by assigning a probability distribution on the heuristic estimates of nodes in G_{js}. To compute the number of nodes expanded, we impose some restrictions on the job processing times and the penalty functions. We first suppose that the processing times A_i are integers and satisfy the condition $1 \leqslant A_i \leqslant k$ for $1 \leqslant i \leqslant N$, for some constant $k > 1$. This imposes a fixed upper bound on the processing times of jobs. We then assume that there is a positive integer constant β such that the penalty $H_i(x)$ is an integer and has order $\mathrm{o}(x^\beta)$ for $1 \leqslant i \leqslant N$. This means that the penalty functions are polynomial in their argument, and the highest power in the polynomial is less than β. We also assume that all arc costs are greater than or equal to 1.

These are reasonable assumptions to make about the processing of jobs in single-machine job sequencing problems. Similar assumptions were made in [27]. To test our assumptions we carried out experiments in which the penalty function for a job was proportional to the square of its completion time. All jobs were submitted at time $t = 0$. The jobs had processing times but no setup times. The objective was to sequence the jobs to minimize the sum of the penalties [29]. Penalty coefficients (proportionality constants) were random integers uniformly distributed in the range 1 to 9. Processing times were random integers uniformly distributed in the range 1 to 99. For a given number of jobs, 100 random problem instances were generated and solved by A*, and the results were averaged over these 100 runs. The heuristic estimate at a node was computed as suggested in [29]. This heuristic is consistent and quite accurate, but we do not know the exact accuracy level, i.e., whether we should take the normalizing function as linear, less-than-linear or logarithmic. The average numbers of nodes generated and expanded by A* on problems of different sizes are shown in Table 1. The number of nodes generated never exceeds $N \cdot Z_t$, where N is the number of jobs. No node is expanded more than once since the heuristic is consistent. The results reported in [27] are similar but slightly better, because an additional pruning rule was applied there. In Table 1, both the number of nodes generated and the number of nodes expanded increase fairly rapidly with the number of jobs.

Let us try to offer a partial theoretical justification for these observations. Let us say that a function $\Psi(N)$ of N varies *exponentially* in N for large N if there exists real constants $\gamma, \delta, \varepsilon > 0$ such that

$$\lim_{N\to\infty} \frac{\Psi(N)}{\delta \cdot \exp(\gamma \cdot N^\varepsilon)} \to 1.$$

Let us also say that $\Psi(N)$ varies *polynomially* in N for large N if there exists a real constant $\varepsilon > 0$ such that

$$\lim_{N\to\infty} \frac{\Psi(N)}{N^\varepsilon} \to 0.$$

We now prove below that when the normalizing function is the identity function and $a_1 < 0$, the expected number of distinct nodes expanded is exponential in N for large N, provided there is a fixed upper bound on the processing times of jobs.

Table 1
Performance of A* using consistent heuristics

No. of jobs	Nodes generated		Nodes expanded	
	Mean	Std. Dev.	Mean	Std. Dev.
18	1299.37	837.37	119.70	86.13
20	2327.02	1523.12	191.52	137.97
22	4207.79	3193.39	313.13	267.38
24	7063.99	5255.22	477.35	389.68
26	12602.69	9273.02	787.99	638.92
28	22046.05	16912.90	1268.68	1044.52
30	37199.94	32578.06	2000.12	1893.12

Theorem 4.1. *Suppose that for $1 \leqslant i \leqslant N$, $H_i(x)$ is $\mathrm{o}(x^{\beta})$ where β is a positive integer constant, and $1 \leqslant A_i \leqslant k$ for some constant $k > 1$. If Φ is the identity function and $a_1 < 0$, then $E(Z_d)$, the expected number of distinct nodes of G_{js} that get expanded, is exponential in N for large N.*

Proof. See Appendix A. □

This theorem is quite general and covers a wide variety of situations that arise in minimum-penalty job sequencing [26]. In particular, the theorem does not assume that the heuristic is admissible, merely that $a_1 < 0$. Some of the constraints can be relaxed as explained below:

(1) There is really no need to assume a constant upper bound on the job processing times. We can just assume that there exists a positive integer constant ε such that A_i is $\mathrm{o}(N^{\varepsilon})$, $1 \leqslant i \leqslant N$.
(2) The proof also remains valid for normalizing functions that are less-than-linear.

We were curious to find out from observation what happens when the heuristic estimates are admissible but not necessarily consistent. So we carried out a further set of experiments making the same assumptions as those made above with a minor change. This time, for a node n, we took $h(n) = h^*(n) - \wp(n)h^*(n)$, where $\wp(n)$ is a uniformly distributed random number in the open interval $(0, 1)$ that is determined independently for each n. The results are shown in Table 2. Since heuristic estimates were inconsistent, a node was expanded more than once as paths of lower cost were found to it. As a result, the number of nodes expanded exceeded the number of nodes generated. These experiments confirm our earlier observation that when the normalizing function Φ is the identity function, the performance of A* tends to deteriorate very fast as N increases.

We were also interested to find out what happens when the heuristic estimates are not purely admissible, i.e., $a_1 < 0$ but $a_2 > 0$. We carried out experiments where heuristic estimates h of nodes were either under-estimated or over-estimated using the formula $h = h^* + \wp h^*$ where $\wp$ is a uniformly distributed random number in the open interval $(-0.5, 0.5)$ that is determined independently for each node. The results obtained can be summarized as follows:

Table 2
Performance of A* under linear error and inconsistent h

No. of	Nodes generated		Nodes expanded	
jobs	Mean	Std. Dev.	Mean	Std. Dev.
6	61.22	4.69	60.55	18.31
8	249.24	22.27	317.28	87.12
10	1001.67	96.65	1606.49	443.89
12	4029.92	400.12	7906.96	1886.63
14	16156.33	1623.79	39349.97	10288.72
16	64752.36	6529.43	185991.06	40743.79

(1) Due to inadmissibility of the heuristic estimate, the number of nodes generated and expanded were reduced, typical values averaged over 100 runs for 16 jobs were 36 034.07 and 17 600.36 respectively. Algorithm A* ran faster than in the admissible case.
(2) The number of node expansions increased by a factor of 4.0 when problem size was increased from 10 to 12 jobs. The factor increased to 4.2 and 4.6 when the number of jobs were increased to 14 and to 16 jobs respectively. Thus the trend indicates that the performance of A* deteriorated faster with problem size as expected.
(3) The solutions obtained are not optimal because the heuristic estimates are sometimes inadmissible.

We restricted our experiments to smaller problems because, for every instance, we needed to find out h^* for all the nodes of the search graph first and store them for computing the heuristic estimates.

What happens when the normalizing function is logarithmic? Unfortunately, nothing very specific can be said.

Example 4.1. Let all arcs in G_{js} have unit cost. This would happen, for example, when the penalty functions H_i take the constant value 1 for all arguments.

(1) Suppose the heuristic estimate function h is perfect, i.e., $h = h^*$ for all nodes. Then Z_d can be linear to exponential in N depending on how ties are resolved.
(2) Now consider the more general case when the heuristic estimates are not necessarily perfect. Let $a_1 < 0$. This means that there is a probability $p > 0$ such that $h(n) < h^*(n)$ with probability p. Hence $f(n) < N$ with probability p. Then for the search graph G_{js}, $E(Z_d) \geqslant 1 + p \cdot C(N,1) + p^2 \cdot C(N,2) + \cdots + p^N = (1+p)^N$ is exponential in N for large N. This is true even when the normalizing function is logarithmic.

Instead of imposing a constant upper bound on arc costs, we can just try to ensure that the majority of the outgoing arcs at a nongoal node have a low cost.

Definition 4.1. Let $\theta : \{1, 2, \ldots, N, \ldots\} \to R$ be a given increasing (total) function with the positive integers as domain and the positive real numbers as range with $\theta(1) = 1$. Let C be a cost function defined on the arcs in G_{js}. Then (G_{js}, C) is *super-regular with respect* *to* θ if for each nongoal node n in G_{js} and for each integer $y > 0$, node n has *at least* $\min\{N - i, \lfloor\theta(y)\rfloor\}$ outgoing arcs with cost $\leqslant y$, where i is the level of node n in G_{js}.

When (G_{js}, C) is super-regular, the arcs costs are said to be θ-relaxed.

Thus super-regular graphs have a large number of outgoing arcs at a node with arc costs lying within a given upper bound, but there is no constant upper bound on the cost of an arc. In this case an exponential number of nodes get expanded when the functions θ and Φ are positive (fractional) powers of their arguments.

Theorem 4.2. *Let $\theta(y) = y^{\beta}$ for $y > 0$, where $0 < \beta \leqslant 1$, and let (G_{js}, C) be super-regular with respect to θ, i.e., θ-relaxed. If $\Phi(x) = \max\{1, x^{\delta}\}$ for $0 < \delta \leqslant 1$, and $a_1 < 0$, then $E(Z_d)$ is exponential in N for large N.*

Proof. See Appendix A. □

4.2. Polynomial complexity

To get polynomial bounds on the number of nodes expanded when

$$\Phi(x) = \max\{1, \ln(x)\},$$

we need to impose an upper bound on the number of arcs emanating from a node with costs lying within a specified limit. This brings us to the notion of a sub-regular graph, which has a definition very similar to that of a super-regular graph, except that instead of a large number of outgoing arcs of bounded cost, we now have a small number of such arcs.

Definition 4.2. Let $\theta : \{1, 2, \ldots, N, \ldots\} \to R$ be a given increasing (total) function with the positive integers as domain and the positive real numbers as range with $\theta(1) = 1$. Let C be a cost function defined on the arcs in G_{js}. Then (G_{js}, C) is *sub-regular* with respect to θ if, for each nongoal node n in G_{js}, the following two conditions are *both* satisfied:

(1) For each integer $y > 0$, there are *at most* $\min\{N - i, \lfloor\theta(y)\rfloor\}$ outgoing arcs from node n with cost $\leqslant y$, where i is the level of node n in G_{js}.
(2) Node n has an outgoing arc of unit cost.

When (G_{js}, C) is sub-regular, the arcs costs are said to be θ-restricted.

An additional assumption about arc costs is also required. Consider the set S_N at the root of G_{js} consisting of the jobs to be sequenced. We can view the elements of S_N as totally ordered under some linear ordering. For example, suppose the jobs in S_N have distinct processing times. Then the jobs can be viewed as ordered in increasing order of processing times. At each node n, the outgoing arcs corresponding to jobs that get processed can be ordered from left to right in order of the processing times of the jobs. We can then impose the condition that the costs of outgoing arcs at node n should also be increasing from left to right, so that the outgoing arc corresponding to the job with the smallest processing time has the lowest cost, and so on.

Definition 4.3. A graph (G_{js}, C) is said to be C-*ordered* if

(i) jobs have distinct processing times, and
(ii) at every nongoal node, whenever the processing time A_i of a job J_i is less than the processing time A_k of a job J_k, the cost of the outgoing arc corresponding to job J_i is less than the cost of the outgoing arc corresponding to job J_k.

In [27], the graphs are C-ordered when the processing times of jobs are distinct and penalty coefficients are identical. C-ordering imposes an additional structure on the graph and enables us to prove the following result when the normalizing function is logarithmic.

Theorem 4.3. *Let $\theta(y) = y^\beta$ for $y > 0$, where $0 < \beta \leqslant 1$, and let (G_{js}, C) be θ-restricted and C-ordered. If Φ is logarithmic, then $E(Z_t)$ is polynomial in N for large N.*

Proof. See Appendix A. □

Our experimental results for this case are given in Table 3. Here the jobs had distinct processing times and the penalty coefficients were identical. Moreover, below every node, there were at most k arcs having arc cost $< 10 \cdot (100k)^2$. Thus these graphs were C-ordered and also θ-restricted except that there might not be an outgoing arc of unit cost below every nongoal node. Heuristic estimates at nodes were admissible and generated using the formula $h = h^* - \wp \log(h^*)$, $\wp$ being defined as in the experiments corresponding to Table 2.

In the next set of experiments, we set $h = h^* - \wp \log(\kappa h^*)$ where κ is a large constant, thereby forcing the logarithmic error to be much larger than for the experiments in Table 3. The results obtained were similar, as shown in Table 4 for $\kappa = 10^{25}$. When the error is logarithmic, the numbers of nodes generated and expanded reduce dramatically. Since the error is relatively small, the heuristic estimate of a node is close to perfect. As a result, when the arc costs at a node are distinct and penalty coefficients are identical, A* generates and expands only a small number of nodes.

The following theorem specifies a sufficient condition which ensures that, in θ-restricted graphs, the total number of nodes expanded by A* is polynomial in N for large N. The required condition imposes an upper bound on the maximum possible value that $\theta(y)$ can attain, which in turn restricts the number of outgoing arcs of bounded cost at any node.

Table 3
Performance of A* with logarithmic error

No. of	Nodes generated		Nodes expanded	
jobs	Mean	Std. Dev.	Mean	Std. Dev.
6	21.00	0.00	5.00	0.00
8	36.00	0.00	7.00	0.00
10	55.00	0.00	9.00	0.00
12	78.00	0.00	11.00	0.00
14	105.00	0.00	13.00	0.00
16	136.00	0.00	15.00	0.00

Table 4
Performance of A* with logarithmic error, $\kappa = 10^{25}$

No. of	Nodes generated		Nodes expanded	
jobs	Mean	Std. Dev.	Mean	Std. Dev.
6	21.04	0.40	5.01	0.10
8	36.06	0.60	7.02	0.20
10	55.16	1.13	9.02	0.14
12	78.10	1.00	11.01	0.10
14	105.24	1.69	13.02	0.14
16	136.55	2.71	15.04	0.20

Theorem 4.4. *Let y_0 be a given positive integer independent of N. Let $\beta > 0$ be an integer constant. Suppose*

$$\theta(y) = \begin{cases} y^{\beta} & \text{for } y \leqslant y_0, \\ y_0^{\beta} & \text{for } y > y_0. \end{cases}$$

Let (G_{js}, C) be θ-restricted and C-ordered. Then regardless of whether Φ is linear, less-than-linear or logarithmic, $E(Z_t)$ is polynomial in N for large N.

Proof. See Appendix A. □

We end this section with the following general result. Here, unlike in the previous theorem, we allow $\theta(y)$ to take a large value for arguments $\geqslant y_0$. In the θ-restricted case, this permits all the remaining arcs at a node to have a cost of y_0.

Theorem 4.5. *Let y_0 be a given positive integer independent of N. Let $\beta > 0$ be a constant. Suppose*

$$\theta(y) = \begin{cases} y^{\beta} & \text{for } y < y_0, \\ N & \text{for } y \geqslant y_0. \end{cases}$$

Then,

(1) *If (G_{js}, C) is super-regular with respect to θ, $\Phi(x) = \max\{1, x^{\delta}\}$ for $0 < \delta \leqslant 1$, and $a_1 < 0$, $E(Z_d)$ is exponential in N for large N;*
(2) *If (G_{js}, C) is sub-regular with respect to θ, G_{js} is C-ordered and Φ is logarithmic, $E(Z_t)$ is polynomial in N for large N.*

Proof. See Appendix A. □

5. Analysis of the Traveling Salesman Problem (TSP)

We now consider the expected complexity of the N-city Traveling Salesman Problem (TSP), assuming the TSP is being solved by means of A* graph search on G_{tsp}. Let cost be an $N \times N$ matrix with positive (nonzero integer) entries that specify the cost (or distance) of travel, $\text{cost}(I, J)$ being the cost of going directly from city I to city J. In the symmetric version of the TSP, cost is a symmetric matrix. We take $\text{cost}(J, J)$, $1 \leqslant J \leqslant N$, to be infinite (i.e., a very large positive integer). We will say that the given cost matrix satisfies the *triangle inequality* if

$$\text{cost}(I, J) + \text{cost}(J, K) \geqslant \text{cost}(I, K)$$

for every triple I, J, K, where $1 \leqslant I, J, K \leqslant N$. This is an intuitively appealing notion, since the usual distance metric in two-dimensional Euclidean space satisfies the triangle inequality.

5.1. Exponential complexity

We now show that when the heuristic is admissible, the normalizing function is the identity function, and the cost matrix is symmetric and satisfies the triangle inequality, the expected number $E(Z_d)$ of distinct nodes expanded is exponential in N for large N. This is a very general result.

Theorem 5.1. *Consider a symmetric N-city Traveling Salesman Problem where the cost matrix satisfies the triangle inequality. Let Φ be the identity function, and suppose $a_1 < 0$. Then $E(Z_d)$ is exponential in N for large N.*

Proof. See Appendix A. □

As can be seen, the heuristic need not really be admissible; a weaker condition suffices. The theorem generalizes readily to normalizing functions that are less-than-linear. The only additional requirement is that the cost of the best tour should not be too large. In the theorem below, we require the cost of the tour to be $O(N)$, but this condition can be relaxed. For example, when $\delta = 0.5$, it suffices if the cost of the best tour is of lower order than N^2.

Theorem 5.2. *Consider a symmetric N-city Traveling Salesman Problem where the cost matrix satisfies the triangle inequality condition. Let $\Phi(x) = \max\{1, x^\delta\}$ for some δ, $0 < \delta < 1$, and let $a_1 < 0$. Then $E(Z_d)$ is exponential in N for large N whenever the cost of the minimum cost solution path is $O(N)$.*

Proof. See Appendix A. □

A similar result can be proved even when the cost matrix is asymmetric, provided $\text{cost}(I, J)$ is not too large compared to $\text{cost}(J, I)$.

Theorem 5.3. *Consider an N-city asymmetric Traveling Salesman Problem where the cost matrix satisfies the triangle inequality condition. Let Φ be the identity function, and let $a_1 < 0$. Suppose there exists a constant ε, $0 < \varepsilon < 1$, such that $\text{cost}(I, J) \leqslant N^\varepsilon \cdot \text{cost}(J, I)$ for $1 \leqslant I, J \leqslant N$, $I \neq J$. Then $E(Z_d)$ is exponential in N for large N.*

Proof. See Appendix A. □

It is possible to relax the constraints even further. The cost matrix does not need to satisfy the triangle inequality condition in a strict way. We will say that the triangle inequality condition holds *α-weakly* if the cost matrix satisfies the following condition for some real number $\alpha > 1$:

$$\alpha \cdot \big(\mathrm{cost}(I, J) + \mathrm{cost}(J, K)\big) \geqslant \mathrm{cost}(I, K)$$

for all triples I, J, K where $1 \leqslant I, J, K \leqslant N$. Theorem 5.1 remains valid when the triangle inequality condition holds α-weakly provided $\alpha = \mathrm{o}(N^{\varepsilon})$ for some $\varepsilon < 1$. Similar conditions can be derived for Theorems 5.2 and 5.3.

In a Traveling Salesman Problem, all pairs of cities may not always be connected directly by arcs, i.e., $\mathrm{cost}(I, J)$ can sometimes be infinite. In such cases, the matrix entries may fail to satisfy the triangle inequality condition as defined above. But even then, a version of Theorem 5.1 holds, provided the following technical condition is satisfied: There is a subsequence of cities $1, 2, \ldots, N^{\varepsilon} + 1$ lying on a minimum cost tour, where $0 < \varepsilon < 1$, such that the cost of the portion of the minimum cost tour lying within the subsequence is $< N^{\varepsilon}$, and the cost matrix restricted to the cities in the subsequence is symmetric and satisfies the triangle inequality.

What can be said when the normalizing function $\Phi(x)$ is logarithmic in x? Unfortunately, it seems that nothing very definite can be said in such a case, as the following examples show. The expected number of node expansions can be linear to exponential in N depending on the cost matrix.

Example 5.1. Suppose $\mathrm{cost}(I, J) = 1$ for all $I, J,\ 1 \leqslant I, J \leqslant N,\ I \neq J$.

This cost matrix is symmetric and satisfies the triangle inequality. Let $a_1 < 0$, i.e., suppose that the heuristic estimates of nodes have a nonzero probability of being admissible. Then, for any nongoal node n, $f(n) < N$ with probability $p > 0$, so that $E(Z_d) = E(Z_t) \geqslant 1 + p \cdot C(N-1, 1) + 2 \cdot p^2 \cdot C(N-1, 2) + \cdots + (N-1) \cdot p^{N-1} = 1 + p \cdot (N-1) \cdot (1+p)^{N-2}$, which is exponential in N for large N. This holds even when the normalizing function is logarithmic.

When the heuristic is perfect, $E(Z_t)$ can be linear to exponential in N depending on how ties are resolved. However, there are cases of polynomial complexity for certain restrictive types of cost matrix as described below.

5.2. Polynomial and nonexponential complexity

We now try to determine what kind of restriction must be placed on the cost function to achieve polynomial complexity when the error is logarithmic.

Theorem 5.4. *Consider a symmetric N-city Traveling Salesman Problem having the following cost matrix*:

$$\mathrm{cost}(I, J) = \begin{cases} a & \textit{when } |I - J| = 1 \textit{ or } (I = 1 \textit{ and } J = N) \textit{ or } (J = 1 \textit{ and } I = N), \\ a + b \cdot \ln N & \textit{otherwise}, \end{cases}$$

where a and b are constants > 0. Suppose $\Phi(x)$ is logarithmic in x. Then the expected number of node expansions $E(Z_t)$ is polynomial in N for large N.

Proof. See Appendix A. □

The conditions in Theorem 5.4 are very restrictive. When two cities I and J are adjacent in the city numbering scheme, then $\text{cost}(I, J)$ is a constant; otherwise, the cost increases logarithmically with N. One way to relax the conditions is as given below, but in that case the result we get is weaker:

Theorem 5.5. *Consider a symmetric N-city Traveling Salesman Problem having the following cost matrix*:

$$\text{cost}(I, J) = \begin{cases} a & \text{when } |I - J| = 1 \text{ or } (I = 1 \text{ and } J = N) \text{ or } (J = 1 \text{ and } I = N), \\ a + b & \text{otherwise}, \end{cases}$$

where a and b are constants $a > b > 0$. Suppose $\Phi(x)$ is logarithmic in x. Then the expected number of node expansions $E(Z_t)$ is not *exponential in N for large N.*

Proof. See Appendix A. □

A similar result is obtained if $\text{cost}(I, J)$ varies with $|I - J|$ at a rate that is no more than linear.

Theorem 5.6. *Let $\Psi(z) = a + b \cdot z^r$, $0 < b < a$ and $0 < r \leqslant 1$. Consider a symmetric N-city Traveling Salesman Problem where for every city I, $1 \leqslant I \leqslant N$, $\text{cost}(I, J) = \Psi(K - 1)$ if $J \equiv (I - 1 \pm K) \bmod N + 1$, K being a positive integer. Suppose Φ is logarithmic in x. Then the total number of node expansions $E(Z_t)$ is* not *exponential in N for large N.*

Proof. See Appendix A. □

The results in this section show that the expected complexity of the A* algorithm for the TSP can be polynomial to exponential in the number of cities, depending on the cost function and the accuracy of the heuristic function. Since A* on graphs can be viewed as a special type of best-first branch-and-bound algorithm, our results here shed some light on the expected complexity of best-first branch-and-bound algorithms for the TSP.

6. Conclusion

This paper has presented a general technique for extending the analysis of the average-case performance of A* from search spaces that are trees to search spaces that are directed acyclic graphs. The topic has importance because many practical problems can be solved more efficiently using search spaces that are graphs rather than trees. We have derived expressions for the expected number of nodes generated by A* and the expected number of node expansions made by A* when it is run on two general types of directed acyclic graphs. Such search graphs are typical of those that arise in certain types of one-machine job sequencing problems. Similar graphs also arise in some solution procedures for the Traveling Salesman Problem (TSP). Our analytical results show that the expected complexity of problems in these two domains can change from exponential to polynomial as the heuristic estimates of nodes become more accurate and restrictions are placed on the cost matrix. We have provided supporting experimental evidence for the one-machine job sequencing problem.

We expect that the analytical approach proposed here can be generalized and extended in a number of directions. One possible way this can be done is by making use of the incremental random tree model described in [14,20,21,31]. Experimentally, we can try to compare the expected number of nodes generated and/or expanded by graph search and tree search on different types of problems. This will tell us how much saving can be realized in computational cost in practice by the use of a search space that is a directed acyclic graph, and can thereby help us to choose between graph and tree search spaces when implementing search algorithms in different application domains.

Acknowledgements

Anup K. Sen was funded by CMDS grant AIT/2117 by the Indian Institute of Management Calcutta. W. Zhang was funded in part by NSF grants IIS-0196057 and ITR/EIA-0113618, and in part by DARPA Cooperative Agreements F30602-00-2-0531 and F33615-01-C-1897. Thanks to the anonymous reviewers for constructive comments.

Appendix A. Proofs

Theorem 4.1. *Suppose that for $1 \leqslant i \leqslant N$, $H_i(x)$ is $\mathrm{o}(x^\beta)$ where β is a positive integer constant, and $1 \leqslant A_i \leqslant k$ for some constant $k > 1$. If Φ is the identity function and $a_1 < 0$, then $E(Z_d)$, the expected number of distinct nodes of G_{js} that get expanded, is exponential in N for large N.*

Proof. The basic idea here is to take an initial segment of a minimal cost solution path in G_{js}, and consider the set S of jobs that get processed in this segment. Each subset of S corresponds to a node in G_{js}. The length of the segment is so chosen that the number of elements in S is a fractional power of N, so that the number of subsets of S is exponential in N. We then show that every such node gets expanded with nonzero probability, and deduce that the expected number of nodes that get expanded is exponential in N.

We renumber jobs if needed and assume that there is a minimal cost solution path in G of cost $M \geqslant N$, such that if we move along it from root to goal, jobs get scheduled from the set $\{1, 2, \ldots, N\}$ in the sequence $(1\ 2 \ldots N)$. Choose $0 < \delta < 1/(\beta + 1)$, and consider the nongoal nodes in G_{js} for which all the missing jobs, corresponding to the jobs already completed, belong to the set $\{1, 2, \ldots, V\}$, where $V = N^\delta$. There are $2^V - 1$ such nodes excluding the root, and for any such node n,

$$g(n) < V \cdot (V \cdot k)^\beta \quad \text{and} \quad h^*(n) > M - V \cdot (V \cdot k)^\beta.$$

Let n' be the node for which the missing elements are exactly $1, 2, \ldots, V$. Since n' lies on a minimum cost solution path, we get $h^*(n') < M$. Moreover, the cost of the path to n' from any predecessor n of n' cannot exceed $V \cdot (V \cdot k)^\beta$. It follows that $M - V \cdot (V \cdot k)^\beta < h^*(n) < M + V \cdot (V \cdot k)^\beta$. Let $a' = a_1/2 < 0$, so that $p = F_\chi(a') > 0$. Then $h(n) \leqslant h^*(n) + a' \cdot \Phi(h^*(n))$ with probability p, so that $f(n) < M$ with probability p, provided $g(n) + h^*(n) < M - a' \cdot \Phi(h^*(n))$. As $a' < 0$ and Φ is the identity function, it suffices to show $M + 2 \cdot V \cdot (V \cdot k)^\beta < M - a' \cdot M + a' \cdot V \cdot (V \cdot k)^\beta$, which always holds for large N because $V^{\beta+1}$ is of smaller order than N, and hence of M. The above conditions are true for n' or any ancestor of n'. Thus if node n is at level i, it gets expanded with probability at least p^i. Therefore, $E(Z_d) \geqslant 1 + p \cdot V + p^2 \cdot C(V, 2) + \cdots + p^V = (1 + p)^V$ which is exponential in N for large N. □

Theorem 4.2. *Let $\theta(y) = y^\beta$ for $y > 0$, where $0 < \beta \leqslant 1$, and let (G_{js}, C) be super-regular with respect to θ. If $\Phi(x) = \max\{1, x^\delta\}$ for $0 < \delta \leqslant 1$, and $a_1 < 0$, then $E(Z_d)$ is exponential in N for large N.*

Proof. Here we make use of the fact that in a super-regular graph there are many outgoing arcs at a node with arc costs lying within a given bound. Since $\theta(1) = 1$ and (G_{js}, C) is super-regular, it follows that each nongoal node in G_{js} has at least one outgoing arc of unit cost. So there is a minimal cost solution path in G_{js} of cost N. Let n be a node at level i. Then $h^*(n) = N - i$. Let $a_1 < a' < 0$, so that $p = F_\chi(a') > 0$. Then $f(n) < N$ with probability p if $g(n) + h^*(n) < N + |a'| \cdot \Phi(h^*(n))$, i.e., if $g(n) < i + |a'| \cdot (N - i)^\delta$. Let us confine our attention to those outgoing arcs having costs $\geqslant 1$ and $< k = 1 + |a'| \cdot ((N - i)^\delta - 1)/i$. All paths from the root to node n composed of such arcs will have their $g(n)$ values within the specified limit. Let P be such a path, and let m be a predecessor of n at level $i' < i$. Then $g(m) < i' + i' \cdot |a'| \cdot (N - i)^\delta / i \leqslant i' + |a'| \cdot (N - i')^\delta$, so $g(m)$ satisfies a similar condition. Thus node n will be expanded with probability $\geqslant p^i$. We will choose $i = N^\rho$, where $0 < \rho < \delta \cdot \beta/(\beta + 1)$. Since there are $i!$ paths from the root to a node at level i, there are at least $\lfloor [\theta(k)]^i / i! \rfloor$ nodes with g-value less than $i + |a'| \cdot (N - i)^\delta$ at level i. Hence $E(Z_d) \geqslant \sum_{j=0}^{N^\rho} p^j \lfloor [\theta(k)]^j / j! \rfloor$. But the right-hand sum is no smaller than $\exp(N^\rho)$ which is exponential in N for large N. □

Theorem 4.3. *Let $\theta(y) = y^\beta$ for $y > 0$, where $0 < \beta \leqslant 1$, and let (G_{js}, C) be θ-restricted and C-ordered. If Φ is logarithmic, then $E(Z_t)$ is polynomial in N for large N.*

Proof. We first find the total number of nodes n for which $g^*(n) + h(n) \leqslant f^*(r) + |a_2| \cdot \ln(f^*(r))$; these include all the nodes that can possibly get expanded. But it is always the case that $h(n) \geqslant h^*(n) - |a_1| \cdot \ln(h^*(n))$. Suppose node n is at level i. Then, since every nongoal node has an outgoing arc of unit cost, we must have $f^*(r) = N$ and $h^*(n) = N - i$. So the condition becomes $g^*(n) < i + k_0 \cdot \ln(k \cdot N)$ where k and k_0 are positive constants. Since G_{js} is θ-restricted, outgoing arcs at a node have costs bounded below by $1, \lfloor 2^{1/\beta} \rfloor, \lfloor 3^{1/\beta} \rfloor, \ldots,$ where $\beta \leqslant 1$, so that $1/\beta \geqslant 1$, implying $\lfloor 2^{1/\beta} \rfloor \geqslant 2$, $\lfloor 3^{1/\beta} \rfloor \geqslant 3$, and so on. In computing upper bounds on the number of expanded nodes, these lower bounds can be viewed as the exact costs of the outgoing arcs. We can now find how many nodes at level i have g^*-values summing up to $\leqslant i + k_0 \cdot \ln(k \cdot N)$.

Since G_{js} is C-ordered, consider the subset of jobs already processed at a node n at level i. Let $J_1, J_2, \dots, J_i$ be the jobs in the subset which are in S_N but are missing from node n. Suppose $J_1 < J_2 < \cdots < J_i$. There are $i!$ ways of scheduling the i jobs leading to $i!$ paths to node n from the root. The scheduling of the jobs in the sequence $J_1, J_2, \dots, J_i$ will determine the path of least cost from the root to node n. Moreover, the completion times of these jobs will be in increasing order. G_{js} being C-ordered, the arc costs in this path will form an increasing ordered sequence of integers, because jobs are processed in increasing order of job number and a penalty function is nondecreasing in its argument. The ordered sequences of arc costs corresponding to the minimal cost paths to two different nodes at level i cannot be identical. Thus we need to count the number of increasing integer sequences of length i which sum up to at most $i + k_0 \cdot \ln(k \cdot N)$. To get an upper bound on the number of such sequences, we take help of the Hardy–Ramanujan asymptotic formula [1, pp. 70, 97] for the number of (unrestricted) partitions of an integer q, which has the form $(A/q) \cdot \exp(B \cdot q^{1/2})$, where A and B are constants. The number of partitions so obtained is polynomial in N. We need to determine the total number of partitions of all integers $\leqslant q$, and this number is also polynomial in N. As explained above, our interest is confined to only those partitions that have $i - 1$ or i parts, where the parts are in increasing order of values. This shows that only a polynomial number of nodes get expanded at each level i, $1 \leqslant i \leqslant N$. Although each node n can get expanded multiple times, $g(n)$ is an integer and can take at most $N + |a_2| \cdot \ln N$ distinct values, so no node gets expanded more than $N + |a_2| \cdot \ln N$ times. It follows that $E(Z_t)$ is polynomial in N for large N. $\square$

Theorem 4.4. *Let y_0 be a given positive integer independent of N. Let $\beta > 0$ be an integer constant. Suppose*

$$\theta(y) = \begin{cases} y^{\beta} & \text{for } y \leqslant y_0, \\ y_0^{\beta} & \text{for } y > y_0. \end{cases}$$

Let (G_{js}, C) be θ-restricted and C-ordered. Then regardless of whether Φ is linear, less-than-linear or logarithmic, $E(Z_t)$ is polynomial in N for large N.

Proof. Let $k_0 = y_0^{\beta}$. Then k_0 is a constant. Every node at level i for $1 \leqslant i \leqslant N$, has at most $\min(k_0, N - i)$ outgoing arcs with arc costs $\leqslant y$ for any $y \geqslant y_0$ and these are the leftmost arcs at that node. Other outgoing arcs can be viewed as having infinite cost. Because of the C-ordering, the jobs that get processed in moving from level 0 to level 1 all have job numbers $\leqslant k_0$; the jobs that get processed in moving from level 1 to level 2 all have job numbers $\leqslant k_0 + 1$; and so on. Thus the number of nodes at level i with finite g^*-value is $\leqslant C(k_0 + i - 1, i)$. So the total number of nodes in the graph with finite g^*-value is less than $\sum_{i=1}^{N} C(k_0 + i - 1, i) + 1 \leqslant C(k_0 + N, N)$ which is polynomial in N for large N. A node n can get expanded only if $g(n) \leqslant f^*(r) \cdot (1 + |a_2|) = N \cdot (1 + |a_2|)$. As in Theorem 4.3, $g(n)$ is an integer and can take at most $N \cdot (1 + |a_2|)$ distinct values, so no node gets expanded more than $N \cdot (1 + |a_2|)$ times. It follows that $E(Z_t)$ is polynomial in N as well. $\square$

Theorem 4.5. *Let y_0 be a given positive integer independent of N. Let $\beta > 0$ be a constant. Suppose*

$$\theta(y) = \begin{cases} y^\beta & \text{for } y < y_0, \\ N & \text{for } y \geqslant y_0. \end{cases}$$

Then,

(1) *If (G_{js}, C) is super-regular with respect to θ, $\Phi(x) = \max\{1, x^\delta\}$ for $0 < \delta \leqslant 1$, and $a_1 < 0$, $E(Z_d)$ is exponential in N for large N;*
(2) *If (G_{js}, C) is sub-regular with respect to θ, G_{js} is C-ordered and Φ is logarithmic, $E(Z_t)$ is polynomial in N for large N.*

Proof. (1) Follows from the proof of Theorem 4.2, since at every nongoal node in the graph, all outgoing arcs have cost $\leqslant y_0$.

(2) Here, since the graph is θ-restricted, at any nongoal node at level i, at most $\min(N - i, \theta(y))$ arcs have cost $\leqslant y$ for $y < y_0$, and at most $N - i$ arcs have cost $\leqslant y$ for $y \geqslant y_0$. Since the graph is also C-ordered, and since the costs of the outgoing arcs are integers, these must be in increasing order of values from left to right, so at any nongoal node there can be at most y outgoing arcs of cost $\leqslant y$. Thus, for $y \geqslant y_0$, (G_{js}, C) is θ-restricted with $\theta(y) = y$. It does not make sense to take $\beta > 1$ in this case. We now relax the given conditions and assume that $\theta(y) = y$ for all y, since this can only increase the expected number of nodes that get expanded. We put $\beta = 1$ and the proof of Theorem 4.3 applies. □

Theorem 5.1. *Consider a symmetric N-city Traveling Salesman Problem where the cost matrix satisfies the triangle inequality. Let Φ be the identity function, and suppose $a_1 < 0$. Then $E(Z_d)$ is exponential in N for large N.*

Proof. Let M be the cost of a minimum cost tour. Without loss of generality, renumber the cities (if necessary) so that $1\ 2 \ldots N\ 1$ is a minimum cost tour, and there is a subsequence of cities $1\ 2 \ldots N^\beta N^\beta + 1$ for some β, $0 < \beta < 1$, such that the cost of the portion of the tour from city 1 to city $N^\beta + 1$ is $\leqslant N^\beta \cdot (M/N)$. Let $k \geqslant 2$ be an integer, and consider a level $i = N^\beta/k$ in G_{tsp}. Let S be the set of nodes at level i with current city $N^\beta + 1$ and all other visited cities belonging to the set $\{1, 2, \ldots, N^\beta\}$. Then a node n in S corresponds to a situation where apart from city 1, $(N^\beta/k) - 1$ cities have been visited out of the cities numbered $2 \ldots N^\beta$. The number of nodes in S is $C(N^\beta - 1, (N^\beta/k) - 1)$, which is exponential in N for large N.

Since $0 > a_1 \geqslant -1$, we have $a' = a_1/2 < 0$. Then $a' > a_1$, so that $p = F_\chi(a') > 0$. We will show that each node n in S will enter OPEN and satisfy $g^*(n) + h(n) < M$ with probability p^i. We will also show that a similar condition will hold for every predecessor of n. Since the minimum cost tour has cost M, this implies that each node n in S will be expanded with probability p^i. For a node n in S we must have $h^*(n) \leqslant$ (cost of traveling from city $N^\beta + 1$ to an unvisited city of lowest number) + (cost of visiting all unvisited cities with numbers in the set $\{2, 3, \ldots, N^\beta\}$ in increasing order of city number) + (cost of visiting the remaining cities in the tour and returning to city 1). Since the cost matrix is symmetric and satisfies the triangle inequality, each of the first and second terms is $\leqslant N^\beta \cdot (M/N)$, and the third term is less than M, so that

$$h^*(n) < M + 2 \cdot N^\beta \cdot (M/N).$$

But,

$$g^*(n) \leqslant N^\beta \cdot (M/N) \quad \text{by the triangle inequality.}$$

Hence,

$$g^*(n) + h^*(n) < M \cdot (1 + 3 \cdot N^\beta / N).$$

On the other hand,

$$h^*(n) \geqslant M - g^*(n) \geqslant M(1 - N^\beta / N).$$

Thus,

$$\begin{aligned} g^*(n) + h(n) &\leqslant g^*(n) + h^*(n) + a' \cdot h^*(n) \quad \text{with probability } p, \\ &< M + a' \cdot M + (3 - a') \cdot M \cdot N^\beta / N \\ &\quad \text{with probability } p, \text{ since } a' < 0, \\ &< M \quad \text{with probability } p, \text{ the third term being ignored since it has} \\ &\quad \text{low order compared to } -a' \cdot M. \end{aligned}$$

For a predecessor n' of n at a level smaller than i in G_{tsp}, $h^*(n)$ has the same bounds, since the arguments remain valid when the current city, instead of being $N^\beta + 1$, is some city in $\{2, 3, \ldots, N^\beta\}$. The bound remains the same for $g^*(n')$ as well, so we still have $g^*(n') + h^*(n') < M$ with probability at least p. We conclude therefore that each node in S at level i, gets expanded with probability at least p^i, so that the expected number of distinct nodes expanded at level i is at least $p^i \cdot C(N^\beta - 1, (N^\beta/k) - 1)$, which for the given value of i is easily shown with the help of Stirling's approximation to be exponential in N for large N if k is chosen to be greater than $1/p$. □

Theorem 5.2. *Consider a symmetric N-city Traveling Salesman Problem where the cost matrix satisfies the triangle inequality condition. Let $\Phi(x) = \max\{1, x^\delta\}$ for some δ, $0 < \delta < 1$, and let $a_1 < 0$. Then $E(Z_d)$ is exponential in N for large N whenever the cost of the minimum cost solution path is* $\mathrm{O}(N)$.

Proof. Similar to the proof of Theorem 5.1. Here we have

$$g^*(n) + h(n) < M + 3 \cdot M \cdot N^\beta / N + a' \cdot M^\delta \cdot (1 - N^\beta / N)^\delta \quad \text{with probability } p$$

where M is $\mathrm{O}(N)$. The parameter β can be chosen as small as we like, and in particular smaller than δ, so that for the given condition on M, the second term can be ignored with respect to the third term, giving $g^*(n) < M$. Indeed, a weaker condition on M suffices; it can be shown that the theorem holds if M is $\mathrm{o}(N^\varepsilon)$ for some $\varepsilon < (1 - \beta)/(1 - \delta)$. □

Theorem 5.3. *Consider an N-city asymmetric Traveling Salesman Problem where the cost matrix satisfies the triangle inequality condition. Let Φ be the identity function, and let $a_1 < 0$. Suppose there exists a constant ε, $0 < \varepsilon < 1$, such that* $\mathrm{cost}(I, J) \leqslant N^{\varepsilon} \cdot \mathrm{cost}(J, I)$ *for $1 \leqslant I, J \leqslant N,\ I \neq J$. Then $E(Z_d)$ is exponential in N for large N.*

Proof. Similar to the proof of Theorem 5.1. Here we have

$$h^*(n) < M \cdot (1 + N^{\beta}/N + N^{\beta+\varepsilon}/N);$$

and the other inequalities remaining the same, so that, with probability p

$$g^*(n) + h(n) < M \cdot (1 + 2 \cdot N^{\beta}/N + N^{\beta+\varepsilon}/N) + a' \cdot M \cdot (1 - N^{\beta}/N).$$

The theorem holds provided we choose β so that $\beta + \varepsilon < 1$. $\square$

Theorem 5.4. *Consider a symmetric N-city Traveling Salesman Problem having the following cost matrix*:

$$\mathrm{cost}(I, J) = \begin{cases} a & \textit{when } |I - J| = 1 \textit{ or } (I = 1 \textit{ and } J = N) \\ & \textit{or } (J = 1 \textit{ and } I = N), \\ a + b \cdot \ln N & \textit{otherwise}, \end{cases}$$

where a and b are constants > 0. Suppose $\Phi(x)$ is logarithmic in x. Then the expected number of node expansions $E(Z_t)$ is polynomial in N for large N.

Proof. Here the cost matrix is symmetric, but the triangle inequality condition is not satisfied. We first find an upper bound on the expected number $E(Z_d)$ of distinct node expansions. The cost of going from a city to the city just before or just after it in the numbering scheme is a constant amount, so $f^*(r) = a \cdot N$. A node n at level i can only get expanded if $g^*(n) + h(n) \leqslant a \cdot N + |a_2| \cdot \ln(a \cdot N)$, i.e., if $g^*(n) + h^*(n) \leqslant a \cdot N + |a_2| \cdot \ln(a \cdot N) + |a_1| \ln(h^*(n))$. Since $h^*(n) \geqslant (N - i) \cdot a$, the required condition for node expansion can be written as $g^*(n) < a \cdot i + a' \cdot \ln N + b'$, where a' and b' are constants. To find an upper bound on $E(Z_d)$, we have to count all nodes n at each level i having $g^*(n) < a \cdot i + a' \cdot \ln N + b'$. Consider a path starting at the root along which nodes are expanded, and examine the order in which cities have been visited and reached. There must exist a positive constant q independent of i such that successive cities have numbers that are adjacent in the city numbering scheme except at a maximum of q positions; otherwise the required condition on the g^*-value cannot be satisfied. Whenever successive cities are not adjacent, there are at most N possible choices for the next city. Therefore the total number of distinct paths from the root to nodes at level i along which node expansions are possible is $< \sum_{j=0}^{q} N^j \cdot C(i, j) < (q + 1) \cdot N^q \cdot C(i, q)$ for large i, and the total number of expansions of nodes in the graph is $< (q+1) \cdot N^q \cdot \sum_{i=0}^{N} C(i, q) < (q+1) \cdot N^{q+1} \cdot C(N, q)$. This is polynomial in N since q is a constant. $\square$

Theorem 5.5. *Consider a symmetric N-city Traveling Salesman Problem having the following cost matrix*:

$$\mathrm{cost}(I, J) = \begin{cases} a & \textit{when } |I - J| = 1 \textit{ or } (I = 1 \textit{ and } J = N) \\ & \textit{or } (J = 1 \textit{ and } I = N), \\ a + b & \textit{otherwise}, \end{cases}$$

where a and b are constants $a > b > 0$. Suppose $\Phi(x)$ is logarithmic in x. Then the expected number of node expansions $E(Z_t)$ is not *exponential in N for large N.*

Proof. We proceed as in the proof of Theorem 5.4. Here the cost matrix is symmetric and also satisfies the triangle inequality condition. For an expanded node n at level i, the upper bound on $g^*(n)$ is the same as in Theorem 5.4. As before, consider a path starting at the root along which nodes are expanded, and examine the order in which cities have been visited and reached. This time there can be at most $\mathrm{O}(\ln N)$ positions where two successive cities are not adjacent in the city numbering scheme. Thus the expected number $E(Z_t)$ of node expansions will be bounded above by an expression of the form $N^{\ln N+1} \cdot C(N, \ln N)$. This bound is not polynomial in N, but nor is it exponential in N. $\square$

Theorem 5.6. *Let $\Psi(z) = a + b \cdot z^r$, $0 < b < a$ and $0 < r \leqslant 1$. Consider a symmetric N-city Traveling Salesman Problem where for every city I, $1 \leqslant I \leqslant N$, $\mathrm{cost}(I, J) = \Psi(K-1)$ if $J \equiv (I - 1 \pm K) \bmod N + 1$, K being a positive integer. Suppose Φ is logarithmic in x. Then the total number of node expansions $E(Z_t)$ is* not *exponential in N for large N.*

Proof. We proceed as in the proof of Theorem 5.4. Here the cost matrix is symmetric and satisfies the triangle inequality condition. For an expanded node n at level i, the upper bound on $g^*(n)$ is the same as in Theorem 5.4. This time, if we take an expanded node and examine the set of visited cities (to which the current city has also been added), there must exist a positive global constant w such that if this set of cities is arranged in increasing order of city number, there can be at most w places where the gap between two successive cities is proportional to $\ln N$. But in addition, there will exist another positive global constant u such that at no more than $\mathrm{O}(\ln N)$ places, the gap between successive cities will be > 1 but $\leqslant u$. The gap between successive cities can take any value between a constant and $\ln N$, so long as the sum of all the gaps is $\mathrm{O}(\ln N)$. Thus to get an upper bound on the number of tours along which nodes can get expanded, we can suppose that there are $\mathrm{O}(\ln N)$ gaps each of which can take $\mathrm{O}(\ln N)$ different values. Thus $E(Z_t)$ will be bounded above by an expression of the form $N \cdot (\ln N)^{\ln N} \cdot C(N, \ln N)$. This bound is not polynomial in N, but nor is it exponential in N. $\square$

References

[1] G.E. Andrews, The Theory of Partitions, in: G.G. Rota (Ed.), Encyclopedia of Mathematics and Its Applications, vol. 2, Addison–Wesley, Reading, MA, 1976.

[2] A. Bagchi, A.K. Sen, Average case analysis of heuristic search in tree-like networks, in: Search in Artificial Intelligence, Springer-Verlag, Berlin/New York, 1988, pp. 131–165.

[3] E. Balas, P. Toth, Branch and bound methods, in: The Traveling Salesman Problem, Wiley, Essex, UK, 1985, pp. 361–401.

[4] M. Bellmore, J.C. Malone, Pathology of traveling-salesman subtour-elimination algorithms, Operations Research 19 (1971) 278–307.

[5] H. Carrillo, D. Lipman, The multiple sequence alignment problem in biology, SIAM J. Appl. Math. 48 (1988) 1073–1082.

[6] S.V. Chenoweth, H.W. Davis, High performance A* search using rapidly growing heuristics, in: Proc. IJCAI-91, Sydney, Australia, 1991, pp. 198–203.

[7] H.W. Davis, Cost error relationships in A∗ tree searching, J. ACM 37 (1990) 195–199.
[8] R. Dechter, J. Pearl, Generalized best-first search strategies and the optimality of A∗, J. ACM 32 (1985) 505–536.
[9] G. Gutin, A.P. Punnen (Eds.), The Traveling Salesman Problem and Its Variations, Kluwer Academic, Dordrecht/Norwell, MA, 2002.
[10] M. Held, R.M. Karp, The traveling salesman problem and minimum spanning trees: Part II, Mathematical Programming 1 (1971) 6–25.
[11] N. Huyn, R. Dechter, J. Pearl, Probabilistic analysis of the complexity of A∗, Artificial Intelligence 15 (1980) 241–254.
[12] T. Ikeda, H. Imai, Enhanced A∗ algorithms for multiple alignments: Optimal alignments for several sequences and k-opt approximate alignments for large cases, Theoretical Computer Science 210 (1999) 341–374.
[13] H. Kaindl, G. Kainz, A. Leeb, H. Smetana, How to use limited memory in heuristic search, in: Proc. IJCAI-95, Montreal, Quebec, 1995, pp. 236–242.
[14] R.M. Karp, J. Pearl, Searching for an optimal path in a tree with random costs, Artificial Intelligence 21 (1983) 99–117.
[15] R.M. Karp, J.M. Steele, Probabilistic analysis of heuristics, in: The Traveling Salesman Problem, Wiley, Essex, UK, 1985, pp. 181–205.
[16] R.E. Korf, M. Reid, S. Edelkamp, Time complexity of iterative-deepening-A∗, Artificial Intelligence 129 (2001) 199–218.
[17] R.E. Korf, W. Zhang, Divide-and-conquer frontier search applied to optimal sequence alignment, in: Proc. AAAI-00, Austin, TX, 2000, pp. 910–916.
[18] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, D.B. Shmoys (Eds.), The Traveling Salesman Problem, Wiley, Essex, UK, 1985.
[19] J.K. Lenstra, A.H.G. Rinnooy Kan, On the expected performance of branch-and-bound algorithms, Operations Research 26 (1978) 347–349.
[20] C.J.H. McDiarmid, Probabilistic analysis of tree search, in: G.R. Gummett, D.J.A. Welsh (Eds.), Disorder in Physical Systems, Oxford Science, 1990, pp. 249–260.
[21] C.J.H. McDiarmid, G.M.A. Provan, An expected-cost analysis of backtracking and non-backtracking algorithms, in: Proc. IJCAI-91, Sydney, Australia, 1991, pp. 172–177.
[22] J. Pearl, Heuristics: Intelligent Search Strategies for Computer Problem Solving, Addison–Wesley, Reading, MA, 1984.
[23] M. Pinedo, Scheduling: Theory, Algorithms and Systems, Prentice Hall, New York, 1995.
[24] A. Reinefeld, T.A. Marsland, Enhanced iterative deepening search, IEEE Trans. Pattern Analysis and Machine Intelligence 16 (1994) 701–710.
[25] T. Sandholm, Algorithm for optimal winner determination in combinatorial auctions, Artificial Intelligence 135 (2002) 1–54.
[26] A.K. Sen, A. Bagchi, Graph search methods for non-order-preserving evaluation functions: Applications to job sequencing problems, Artificial Intelligence 86 (1996) 43–73.
[27] A.K. Sen, A. Bagchi, R. Ramaswamy, Searching graphs with A∗: Applications to job sequencing, IEEE Trans. Syst. Man Cybern. A 26 (1996) 168–173.
[28] A.K. Sen, A. Bagchi, W. Zhang, An average-case analysis of graph search, in: Proc. AAAI-02, Edmonton, Alberta, 2002, pp. 757–762.
[29] W. Townsend, The single machine problem with quadratic penalty function of completion times: A branch and bound solution, Management Science 24 (1978) 530–534.
[30] W. Zhang, Depth-first branch-and-bound versus local search: A case study, in: Proc. AAAI-00, Austin, TX, 2000, pp. 930–936.
[31] W. Zhang, R.E. Korf, Performance of linear-space search algorithms, Artificial Intelligence 79 (1995) 241–292.