[PST91] S.A. Plotkin, D.B. Shmoys, and É. Tardos. Fast approximation algorithms for fractional packing and covering problems. *Proceedings 32nd Annual Symposium on Foundations of Computer Science*, pages 495–504, 1991.

[PY91] C.H. Papadimitriou and M. Yannakakis. Shortest paths without a map. *Theoretical Computer Science*, 84(1):127–50, July 1991.

[PY93] C.H. Papadimitriou and M. Yannakakis. The traveling salesman problem with distances one and two. *Mathematics of Operations Research*, 18(1):1–11, February 1993.

[ST85] D.D. Sleator and R.E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–8, February 1985.

[Tar95] É. Tardos. *Private communication*, 1995.

[Wes95] D. West. *Private communication*, 1995.

[Wil93] T. G. Will. *Extremal Results and Algorithms for Degree Sequences of Graphs*. PhD thesis, U. of Illinois at Urbana-Champaign, 1993.

Lastly, is there a polynomial-time approximation scheme for the trp for trees with general distributions? We conjecture that at least a *pseudo-polynomial* time approximation scheme exists. If so, polyhedral separation and duality could imply that the randomized search ratio problem is so approximable in trees.

## References


[ACP+86] F. Afrati, S. Cosmadakis, C.H. Papadimitriou, G. Papageorgiou, and N. Papakostantinou. The complexity of the travelling repairman problem. *Informatique Theórique et Applications*, 20(1):79–87, 1986.

[BCR88] R.A. Baeza-Yates, J.C. Culberson, and G.J.E. Rawlins. Searching with uncertainty. *SWAT 88. 1st Scandinavian Workshop on Algorithm Theory. Proceedings*, pages 176–89, 1988.

[BCC+94] A. Blum, P. Chalasani, D. Coppersmith, W. Pulleyblank, P. Raghavan, and M. Sudan. The minimum latency problem. *Proceedings 26th Annual Symposium on Theory of Computing*, pages 163–171, 1994.

[Chr76] N. Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. *Technical report, GSIA, Carnegie-Mellon University*, 1976.

[DKP91] X. Deng, T. Kameda, and C. Papadimitriou. How to learn an unknown environment. *Proceedings 32nd Annual Symposium on Foundations of Computer Science*, pages 298–303, 1991.

[DP90] X. Deng and C.H. Papadimitriou. Exploring an unknown graph. *Proceedings 31st Annual Symposium on Foundations of Computer Science*, pages 355–361 vol. 1, 1990.

[FLM93] M. Fischetti, G. Laporte, and M. Martello. The delivery man problem and cumulative matroids. *Operations Research*, vol. 41, pages 1055–1064, 1993.

[GK96] M. Goemans and J. Kleinberg. An improved approximation ratio for the minimum latency problem. *Proceedings Annual Symposium on Discrete Algorithms*, to appear, 1996.

[GLS88] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization.* Springer-Verlag, 1988.

[Min89] E. Minieka. The delivery man problem on a tree network. *Annals of Operations Research*, vol. 18, pages 261–266, 1989.

Let $\ell_1$ be the total length of $E_1$; it can be shown that the total trp cost of the smallest among $S_1$ and $S_1^r$ is bounded from above by $\frac{\ell_1 \cdot n}{2}$ (each node is first visited on the average with delay $\frac{\ell_1}{2}$ or better in the two traversals). Next, let $\ell_2$ be the total length of $E_2$; it is easy to see that the total trp cost of $S_2$ is bounded from above by $\ell_2 \cdot n - \frac{n^2}{2} + o(n^2)$.

The key observation now is that the sum of $\ell_1$ and $\ell_2$ is at most $(3+\delta) \cdot n$, because the shortest walk from $r$ contains *both* a matching of the odd-degree nodes, *and* a matching of the odd-degree nodes minus $r$. Hence, the performance of the algorithm is bounded above by

$$\max_{\substack{\ell_1+\ell_2 \leq (3+\delta)\cdot n \\ 0 \leq \delta \leq 1}} \frac{\min\{\frac{\ell_1 \cdot n}{2}, \ell_2 \cdot n - \frac{n^2}{2}\}}{\frac{1}{2}(1+\delta^2)\cdot n^2}.$$

It turns out that this expression evaluates to $\frac{5+\sqrt{29}}{6} \approx 1.73$

To improve this to $1.6615\ldots$ we must argue that the total trp cost is bounded from above by $2\ell_2 n - \frac{1}{2}\ell_2^2 - n^2 + o(n^2)$, instead of the more pessimistic $\ell_2 \cdot n - \frac{n^2}{2} + o(n^2)$. The argument involves decomposing $E_2$ into a path and several Eulerian graphs, and choosing for each Eulerian graph the better of two traversals. The expression now becomes

$$\max_{\substack{\ell_1+\ell_2 \leq (3+\delta)\cdot n \\ 0 \leq \delta \leq 1}} \frac{\min\{\frac{\ell_1 \cdot n}{2}, 2\ell_2 n - \frac{1}{2}\ell_2^2 - n^2\}}{\frac{1}{2}(1+\delta^2)\cdot n^2},$$

and it evaluates to $\max_{x\in[1,2]} \frac{4x-x^2-2}{1+(x^2-5x+5)^2} \approx 1.6615$, which is maximized when $x \approx 1.4545$ is the unique root between 1 and 2 of the equation $x^5 - 11x^4 + 44x^3 - 75x^2 + 44x + 2$. $\square$

## 4 Open Problems

Can we achieve in polynomial time better approximations than those in Theorem 2? Naturally, we can do better for graphs for which the TSP is solvable exactly, or has a better approximation ratio than $\frac{3}{2}$, as in the case of TSP with distances 1 and 2 [PY93]. Also, our approximation of $\sigma(G,r)$ can be extended to *weighted* graphs. Can our approximation of $\rho(G,r)$ be also so extended?

Computing $\sigma(G,r)$ and $\rho(G,r)$ when $G$ is a tree is a surprisingly tough problem. An NP-completeness proof for the tree case (not unlikely, in view of the many NP-complete mean-flow scheduling problems with a similar flavor) would establish the NP-hardness, via duality and polyhedral separation, of the trp for trees (a problem long conjectured to be NP-complete).

Can we improve the approximation ratio for the uniform trp to 1.5? Further, in the nonuniform (weighted) case the current ratio is still rather large.

**Corollary 1** *The search ratio problem and the randomized search ratio problem can be solved exactly in polynomial time for trees with a bounded number of leaves.*

The work of [PST91, Tar95] and [GLS88] suggest that any polynomial time *approximation scheme* for the trp can be transferred to the randomized search ratio problem in the same class of graphs.

### 3.2 Approximation of the Uniform trp

We have a graph $G$, with a fixed root $r$, and we wish to find a walk starting from $r$ that visits all nodes, and minimizes the sum of the arrival times at the nodes. It is easy to see that an approximation ratio of 2 can be achieved by a simple spanning tree heuristic that traverses a spanning tree in depth-first order. Formulating and analysing the analogue of Christofides' algorithm in this setting is nontrivial. Our approximation algorithm is the following:

Find a spanning tree $T$ of $G$ in which $r$ has odd degree.
Find a shortest matching $M_1$ of all odd-degree nodes of $T$.
Add $M_1$ to $T$, to obtain an Eulerian graph $E_1$.
Find a traversal of $E_1$, and its reverse, call them $S_1$ and $S_1^r$.

Find a shortest matching $M_2$ of all odd-degree nodes of $T$ *except* for $r$ and some other node.
Add $M_2$ to $T$ to obtain an *almost* Eulerian graph $E_2$ with two odd-degree nodes.
Find a traversal of $E_2$, call it $S_2$.

Select the best among $S_1$, $S_1^r$, and $S_2$.

The first line is impossible if $r$ is an articulation point belonging to an even number of components, to all as a leaf; in this case we add a new node $r'$ adjacent only to $r$, and call it the root; the performance is not affected.

**Theorem 7** *The algorithm above yields a solution to the trp which is at most* 1.662 *times the optimum.*

The precise ratio in the statement of the theorem is $\max_{x\in[1,2]} \frac{4x-x^2-2}{1+(x^2-5x+5)^2} \approx 1.6615$, which is maximized when $x \approx 1.4545$ is the (unique) root between 1 and 2 of the polynomial $x^5 - 11x^4 + 44x^3 - 75x^2 + 44x + 2$.

**Proof.** Suppose that the length of the shortest walk in $G$ starting from $r$ and visiting all nodes is $n(1+\delta)-1$, for some $\delta$ between 0 and 1. It can be shown (proof omitted) that $\frac{1}{2}n^2(1+\delta^2)+o(n^2)$ is a *lower bound* on the trp (otherwise, we would be able to find a shorter walk from $r$).

**Theorem 3** *If the distances in the metric $d$ are polynomially small integers, and the probabilities* **pr** *are rational numbers with small coefficients and common denominators, then the two problems are polynomially equivalent.*

**Proof.** In one direction we simulate distances by long paths whose intermediate nodes have zero probabilities; in the other we simulate a node with probability $\frac{A}{B}$, where $B$ is the common denominator, by a cluster of $A$ nodes with distance zero from one another. $\square$

We shall henceforth focus on the graph version of the trp.

**Theorem 4** *The trp with the uniform distribution is NP-complete (and MAXSNP-hard).*

**Proof.** Another easy reduction from Hamilton path, omitted. $\square$

Can we solve the trp exactly on any interesting class of graphs? It follows from the results of [ACP$^+$86] that it can be solved on paths. We can generalize this a little:

**Theorem 5** *On trees with $L$ leaves, the trp can be solved in $O(n^L)$ time.*

**Proof.** Such a tree has $O(n^L)$ subtrees; furthermore, the optimum trp solution is guaranteed to end up in a leaf, and thus dynamic programming is enabled. $\square$

It is worth mentioning that the trp with the uniform distribution is solvable for general trees; in fact, any depth-first traversal is optimal.

### 3.1 Polyhedral Separation

Suppose that we wish to solve the linear programming formulation LP of the randomized search ratio problem. In fact, we should solve the dual, which has manageably many dimensions:

$$\begin{aligned} &\min \sum_v d(r,v)y_v - z \\ &\sum_v d_\pi(r,v)y_v - z \geq \quad 0 \quad (\text{all } \pi) \qquad\qquad (\text{DLP}) \\ &\qquad\qquad y_v \geq 0 \end{aligned}$$

Suppose then that we wish to solve DLP by the ellipsoid algorithm [GLS88]. We are given a point $(\bar{y}, z) \in \Re^{n+1}$, and we are asking whether or not it lies within the feasible region of DLP; if not, we need a violated inequality. It is easy to see that *this is precisely the trp problem.* Hence we have:

**Theorem 6** *If the trp can be solved in polynomial time for a class of graphs* and any distribution*, then the randomized search ratio problem can be solved exactly in polynomial time for that class of graphs.*

Christofides algorithm [Chr76]. Let $\chi_k$ be the length of the tour computed by Christofides algorithm for exploring $S_k$. Notice that the optimum tour that visits all nodes in $S_k$ is at most $p_k + k$ (the result of visiting all nodes using an optimal path and then returning to the root). It follows that $\chi_k \leq 1.5(p_k + k) \leq 1.5(r+1)k$. Hence, this simple modification of the doubling heuristic finds a ratio that is at most $6(\sigma + 1)$ (notice the additive constant, whence the "asymptotic" in the statement of the theorem).

For the randomized version, Christofides' algorithm gives an approximation ratio of $6 + 2\sqrt{10} \approx 12.35$: The lower bound is now $\frac{1}{2}|S_k|(1 + (\frac{p_k - |S_k|}{|S_k|})^2)$ (proof omitted), whereas Christofides' algorithm gives a path of length at most $\frac{3|S_k|}{2} + \frac{p_k - |S_k|}{2}$; the worst-case ratio of the two is $6 + 2\sqrt{10} \approx 12.35$.

But we can do better by a rather novel kind of randomization. Our approximation algorithm is still deterministic, but the solution it produces will in fact be a *distribution* of walks on $G$. In particular, suppose that the tours produced by Christofides' algorithm in the various stages are $T_1, T_2, \ldots, T_m$. The distribution we produce selects the tour $(T_1^{e_1}, \ldots, T_m^{e_m})$ with probability $\frac{1}{2^m}$, where the $e_i$'s are either 1 or $-1$, denoting possible reversal of the tour. In other words, at each phase we try both the tour and its reverse, with equal probability. As a result, the target node is expected to be encountered at the middle of the last tour. Hence the ratios to be minimized now become $\frac{x_1 + x_2 + \cdots + x_{n-1} + \frac{1}{2}x_n}{x_{n-1}}$. The recurrence for the $x_i$'s is now $x_n = (2r-1)x_{n-1} - 2rx_{n-2}$, which is feasible (the corresponding algebraic equation has real roots) only when $4r^2 - 12r + 1 \geq 0$, or $r \geq \frac{3}{2} + \sqrt{2}$. Hence the radius in the "doubling heuristic" now is increased by factors of $1 + \sqrt{2}$, and the approximation ratio becomes $\frac{3+\sqrt{10}}{2} \cdot \frac{3+2\sqrt{2}}{2} \approx 8.98$. □

## 3 The Traveling Repairman Problem

The trp has been originally defined [ACP+86] on an arbitrary (non-graph) metric $d$ on $n$ points, where we seek to

$$\min_{\pi} \sum_{i=1}^{n} \sum_{j=1}^{i-1} d_{\pi(j),\pi(j+1)},$$

where $\pi(1) = r$. Here we define it on a graph $G$, with an arbitrary distribution **pr** on the nodes, and a root $r$, where we must

$$\min_{\pi} \sum_{v \in G} \mathbf{pr}(v) d_{\pi}(r, v),$$

where $\pi$ now ranges over all walks of the graph. It is not hard to observe that, under very mild restrictions (not affecting, for example, approximability) these two versions are equivalent:

## 2.1 Approximation

**Theorem 2** *There are polynomial-time approximation algorithms for the search ratio and randomized search ratio problems with* asymptotic *approximation ratio* 6 *and* $\frac{3+\sqrt{10}}{2} \cdot \frac{3+2\sqrt{2}}{2} \approx 8.98$, *respectively.*

**Proof.** Consider the following family of heuristics:

> **for** $i := 0$ **to** $m$ **do**
> Let $G_i$ be the graph $G$ restricted to all nodes with distance $x_i$ or less from $r$; (* *comment:* $x_m$ is the radius of $G$ *)
> search $G_i$ depth-first, and return to $r$.

Let $S_k$ be the set of nodes in distance $k$ or less from $r$. Then it is easy to see that the search ratio $\sigma$ is at least $\frac{|S_k|-1}{k}$. In phase $i$, it takes $2(|S_{x_i}|-1)$ steps to explore all nodes and return to $r$. If the target node is found during the $n$-th phase, the on-line cost is at most $2(|S_{x_1}|-1)+2(|S_{x_2}|-1)+\ldots+2(|S_{x_n}|-1)$. Since the optimum is at least $x_{n-1}$ the search ratio is at most

$$\frac{2(|S_{x_1}|-1)+2(|S_{x_2}|-1)+\ldots+2(|S_{x_n}|-1)}{x_{n-1}} \leq \frac{2\sigma x_1 + 2\sigma x_2 + \ldots 2\sigma x_n}{x_{n-1}}$$

The optimal strategy is to choose $x_i = 2^i$. (Simple proof: We want to minimize $\max_n \frac{x_1+\ldots+x_n}{x_{n-1}}$. Notice that all fractions except for the $n$th are nondecreasing in $x_n$, and the $n$th is decreasing; it follows that at minimax they are all equal. Call this value $r$. To solve for $x_n$ we have $x_n = r(x_{n-1}-x_{n-2})$, which gives increasing $x_i$'s only if $r \geq 4$. Adopting this minimum value gives $x_i$s that are powers of 2, up to a constant. End of proof that powers of two are optimal.) The *doubling heuristic*, with ratio at most $8\sigma$, results.

For the randomized search ratio, let $u$ be a random node in $S_k$. Any deterministic on-line algorithm will explore on the average $(|S_k|-1)/2$ nodes before $u$. Since a randomized algorithm is simply a distribution of deterministic algorithms the same holds for randomized algorithms. Therefore, for any randomized algorithm there exists a node $u$ in $S_k$ that is expected to be reached after $(|S_k|-1)/2$ steps. This gives a lower bound $\frac{|S_k|-1}{2k}$ of the randomized search ratio —half the lower bound of the deterministic search ratio. It follows that the doubling heuristic produces a search strategy which, seen as a distribution, has expected ratio at most 16 times the optimum randomized search ratio.

The above lower bound $\frac{|S_k|-1}{k}$ (or $\frac{|S_k|-1}{2k}$) of the optimum search ratio is too crude. A better lower bound results by improving the numerator: Any on-line algorithm needs at least $p_k$ steps to explore all nodes in $S_k$, where $p_k$ is the length of the minimum TSP path. Therefore, the optimum search ratio $r$ is at least $\frac{p_k}{k}$. Hence, instead of exploring the nodes in $S_{x_i}$ in a depth-first manner, we can use three problems are polynomial-time solvable in this case). In fact, the techniques in [PST91, Tar95] and [GLS88] suggest that a *polynomial-time approximation scheme* for the trp may be transferable to the randomized search ratio problem (for the same class of graphs).

The trp can be approximated within a constant factor: [BCC+94] gives an algorithm with approximation ratio 144, and [GK96] improves this to 21.55. It is possible to do better in the case of *uniform* distribution. A simple spanning tree heuristic achieves ratio 2. We give an interesting variant of Christofides' algorithm [Chr76] for the trp with *uniform* distribution, and show that it achieves approximation ratio 1.662.

## 2 Computing the Search Ratio

Both versions of the search ratio problem have been defined in the introduction.

**Theorem 1** *Computing the search ratio and the randomized search ratio of a graph $G$ with respect to a root node $r$ is NP-complete and MAXSNP-hard.*

**Proof.** Both are easy reductions from the Hamilton path problem. Given a graph $H$, we define $G$ as $H$ plus a new node $r$, and edges from $r$ to all vertices of $H$. It is easy to see that $\sigma(G,r)$ is $h$ or less and $\rho(G,r)$ is $h/2$ or less, where $h$ is the number of nodes of $H$, if and only if there was a Hamilton path in $H$. MAXSNP-hardness for the search ratio follows as well. MAXSNP-hardness for the randomized version is a little trickier, because this reduction is *not* an L-reduction. However, it can be shown that, if the smallest Hamilton walk of $H$ is $h(1+\epsilon)$, then the randomized search ratio of $G$ is at least $\frac{1}{2}h(1+\epsilon^2)$, and this suffices.

It is a little more nontrivial to argue that the randomized search ratio problem is in NP. It is not hard to verify that $\rho(G,r)$ can be reformulated as follows:

$$\begin{array}{rcll} \min & \rho & & \\ \sum_\pi x_\pi d_\pi(r,v) & \leq & \rho \cdot d(r,v) \quad (\text{all } v) & \qquad (\text{LP}) \\ \sum_\pi x_\pi & = & 1 & \\ x_\pi & \geq & 0 & \end{array}$$

This is an $(n+1) \times n!$ linear program. However, the optimum value will be a basic feasible solution having at most $n+1$ walks with nonzero probability. Such a solution can be guessed, computed, and compared with any given bound, establishing that the problem is in NP. □

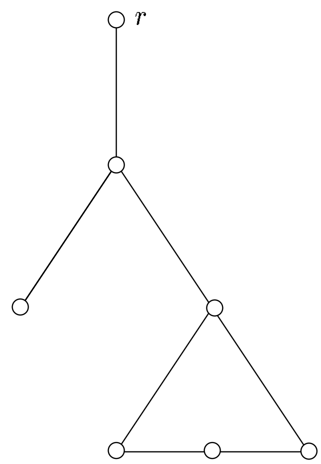

Figure 1: A graph $G$ with $\sigma(G,r)=\frac{7}{3}$ and $\rho(G,r)=2$.

optimize the expected cost. That is, we would be trying to

$$\min_{\pi} \sum_{v\in G} \mathbf{pr}(v) d_{\pi}(r,v).$$

Interestingly, this is an equivalent formulation of a rather well-studied and notoriously hard problem, the *traveling repairman problem (trp)*[3] [ACP+86, Wes95, Wil93]: Given a metric and a starting point, find the route that minimizes the sum of the arrival times at the points. It is known only how to solve on paths [ACP+86] —even the case of *trees*, even that of *caterpillars* (paths with edges sticking out), is conjectured to be hard [Wes95]. We observe that the problem can be solved in polynomial time (for any distribution) for a class of graphs slightly more general than paths: trees with a bounded number of leaves.

We also point out something rather unexpected: *The trp is the polyhedral separation problem of the dual of the randomized search ratio problem.* That is to say, if we can solve the trp for some metric, and for arbitrary distribution, then we would be able to solve the dual of the randomized search ratio problem (and thus the randomized search problem itself) for the same metric by using the ellipsoid algorithm [GLS88]; unfortunately, as we mentioned above we can only solve it in the fairly restricted case of trees with bounded number of leaves (it follows that all

[3]This problem has been studied also under the names *delivery man* [FLM93, Min89] and *minimum latency* [BCC+94, GK96]. [Wil93] calls this the *school-bus driver problem*, with this amusing explanation: A bus driver tries to deliver the children in his/her bus so as to minimize not travel time, but time weighted by the number of children (and ensuing havoc) in the bus...

is an on-line problem of a rather familiar genre: *exploration and navigation* [PY91, DP90, DKP91]. However, unlike previous formulations of such on-line problems, here we *know* the terrain[1] being explored. In other words, for each graph $G$ and start (root) node $r$ of $G$ there is an optimal competitive ratio,

$$\sigma(G, r) = \min_{\pi} \max_{v \in G} \frac{d_\pi(r, v)}{d(r, v)}.$$

Here $\pi$ ranges over all walks of $G$, starting from $r$, and visiting all nodes of $G$, while $d_\pi(r, v)$ is the distance traversed in walk $\pi$ until we first visit node $v$. We call $\sigma(G, r)$ the *search ratio* of the graph with respect to the root. It is an interesting graph-theoretic parameter of a rather novel kind. Unfortunately, we point out (Theorem 1) that it is NP-complete to compute —in fact, our proof establishes that it is MAXSNP-complete.

We may of course want to introduce the *randomized version* of the search ratio:

$$\rho(G, r) = \min_{\Delta} \max_{v \in G} \frac{\mathcal{E}_\Delta[d_\pi(r, v)]}{d(r, v)},$$

where $\Delta$ ranges over *distributions of walks*. We call this parameter the *randomized search ratio*. For example, for the graph and root shown in Figure 1, the search ratio is $\frac{7}{3}$, while the randomized search ratio is 2. Computing $\rho(G, r)$ is also NP-complete; in fact, the surprising part here is that it is in NP at all —we establish this in Theorem 1 via a *linear programming formulation* of the problem. It is also MAXSNP-hard, although this too is somewhat tricky to establish.

We present polynomial-time algorithms for approximating these parameters within a fixed factor. A simple *doubling heuristic* (repeatedly double the radius, explore the resulting graph depth-first) achieves an approximation ratio of 8 for the deterministic ratio, 16 for the randomized version.[2] We improve on this basic algorithm in several directions: By using Christofides algorithm for traversing the graph we improve the guarantees to 6 and 12.35, respectively. By using a novel kind of randomization (and an expansion factor other than two) we improve the latter to 8.98.

Competitive analysis is supposed to be a novel alternative to the classical approach to decision-making under uncertainty: *expectation minimization.* In expectation minimization we would assume a distribution for the node sought, and

[1]An example of previous work on searching a known terrain is the *bridge problem*, sometimes called the *cow path problem*, [BCR88]; our work can be seen as a generalization of this problem from infinite paths to general graphs.

[2]Incidentally, notice the novelty of the situation: We approximate within a bounded ratio a parameter which is *itself* a ratio of a feasible solution divided by an ideal solution! That is, we mix two well-studied compromises: one in the face of uncertainty, the other in the face of computational complexity. Complexity issues are traditionally ignored in the context of competitive analysis.

# Searching a Fixed Graph

Elias Koutsoupias* elias@cs.ucla.edu

Christos Papadimitriou† christos@cs.ucsd.edu

Mihalis Yannakakis‡ mihalis@research.att.com

November 10, 1995

**Abstract**

We study three combinatorial optimization problems related to searching a graph that is known in advance, for an item that resides at an unknown node. The *search ratio* of a graph is the optimum *competitive ratio* (the worst-case ratio of the distance traveled before the unknown node is visited, over the distance between the node and a fixed root, minimized over all Hamiltonian walks of the graph). We also define the *randomized search ratio* (we minimize over all *distributions* of permutations). Finally, the *traveling repairman problem* seeks to minimize the expected time of visit to the unknown node, given some distribution on the nodes. All three of these novel graph-theoretic parameters are NP-complete —and MAXSNP-hard— to compute exactly; we present interesting approximation algorithms for each. We also show that the randomized search ratio and the traveling repairman problem are related via *duality* and *polyhedral separation*.

## 1 Introduction

Imagine that you know that an information item you need resides at *some* node of a fixed graph (say, a large network of hypertext documents), but you do not know where. You can only navigate the graph by following its edges, at unit cost (that is, we assume that there is no random access of pages). You will see the item once you arrive at the right node —and only then. What are good strategies for performing this task efficiently?

This is obviously a situation of decision-making under uncertainty, and therefore an invitation for applying the techniques of *competitive analysis* [ST85]. In fact, this

*CS Department, UCLA.

†CSE Department, UCSD. Research supported in part by a NSF grant.

‡AT&T Bell Laboratories, Murray Hill.