The American Mathematical Monthly

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/uamm20

# A Jeep Crossing a Desert of Unknown Width

Richard E. Korf

**To cite this article:** Richard E. Korf (2022) A Jeep Crossing a Desert of Unknown Width, The American Mathematical Monthly, 129:5, 435-444, DOI: 10.1080/00029890.2022.2051404

**To link to this article:** https://doi.org/10.1080/00029890.2022.2051404

Published online: 28 Apr 2022.

# A Jeep Crossing a Desert of Unknown Width

**Richard E. Korf**

**Abstract.** The classic jeep problem concerns crossing a desert wider than the range of the jeep, with the aid of preplaced fuel caches. There has been a lot of work on this problem and its variations, and the optimal strategy is well known, but all previous work assumes that we know the width of the desert. We consider the case where we don't know the distance in advance. We evaluate a strategy by its competitive ratio, which is the worst-case ratio of the cost of the strategy, divided by the cost of an optimal solution had we known the distance in advance. We show that no strategy with a fixed sequence of caches can achieve a finite competitive ratio. The optimal strategy is an iterative one that uses the optimal known-distance strategy to reach a sequence of target distances, emptying all caches between iterations. An optimal iterative strategy doubles the cost of each successive iteration, and achieves a competitive ratio of four.

**1. INTRODUCTION.** We have a jeep with a given fuel capacity, and a range that it can travel on a full load of fuel. Without loss of generality, we assume that the jeep can carry one gallon of fuel, and can travel one mile per gallon, but want to cross a desert more than a mile wide. We have an unlimited amount of fuel at the start, and can cache unlimited fuel along the way. We want to minimize the cost to travel a given distance, which we measure as the fuel taken from the start. This is an upper bound on the total distance traveled or fuel consumed, if any fuel is left behind in the caches.

In the one-way version of the problem, we just have to cross the desert, while in the two-way version we have to return to the start as well. We adopt the two-way version, since we don't know the distance, and always have to be able to return to the start.

**2. PREVIOUS WORK.** The first solutions to this problem were provided by [3] and [7] in 1947. It is much easier to determine the maximum distance we can travel on a given amount of fuel than the minimum fuel needed to travel a given distance. [3] solved the one-way problem, and [7] solved the round-trip problem. [7] also introduced an alternative version with a caravan of jeeps that can share fuel, only one of which has to complete the trip. Other versions of this problem appear in [1, 2, 4–6].

With one gallon of fuel, we can go $1/2$ mile and return. With two gallons, we go $1/4$ mile, cache $1/2$ gallon, and return. Then we go $1/4$ mile, pick up $1/4$ gallon, go $1/2$ mile further and back, pick up the remaining $1/4$ gallon, and return to the start. With three gallons, we can go $1/6 + 1/4 + 1/2$ miles, with caches at $1/6$ and $1/6 + 1/4$ miles from the start. In general, with $n$ gallons, we can go $1/2 + 1/4 + 1/6 + \cdots + 1/2n$ miles, with $n - 1$ caches, spaced at these distances from each other, working back from the end. We never carry back more fuel than needed to reach the previous cache or start, and every forward hop starts with a full gallon of fuel.

To find the minimum amount of fuel needed to travel a given distance, we find the largest sum of this series that is less than or equal to the given distance, and place the caches at those locations working back from the end. Then we make as many trips as needed from the start to transfer the requisite amount of fuel to the first cache.

The amount of fuel needed to go $d$ miles and return is $O\left(e^{2d}\right)$ gallons [7]. Thus, the fuel cost is proportional to approximately $7.389^d$ gallons, in the limit of large $d$.

doi.org/10.1080/00029890.2022.2051404
MSC: Primary 00A08, Secondary 90B06; 68W40

**3. UNKNOWN DISTANCE AND COMPETITIVE RATIO.** All of the previous work on this problem assumes that we know how far we need to go. We consider here the optimal strategy when we don't know in advance how far we have to travel.

To evaluate such a strategy, we use its competitive ratio [8]. This technique is often used to analyze an algorithm when information critical to its performance is not known a priori. The competitive ratio of an algorithm for this problem is the cost of reaching a given distance using the algorithm, divided by the cost of an optimal solution for that distance, had we known the distance in advance. The worst-case competitive ratio is the maximum value of this ratio over all possible distances, and an optimal algorithm is one with the smallest such maximum value.

**4. OVERVIEW.** We first consider the case of fixed, evenly spaced caches, and show that no such strategy can achieve a finite competitive ratio. We then show that any solution with a fixed sequence of caches can be transformed into one with evenly spaced caches without any loss of efficiency. We then introduce an iterative strategy where one travels a sequence of successively longer distances, using the known optimal strategy for each distance, and empties all the caches between iterations. Finally, we show that one optimal sequence of iterations doubles the cost of each preceding iteration, and increases the distance by a little over a third of a mile with each iteration.

**5. STRATEGIES WITH A FIXED SEQUENCE OF CACHES.** The simplest strategy would repeatedly stock a fixed sequence of caches, until we reach the goal.

To reach the goal as soon as possible, we always want to go forward as far as possible, without sacrificing efficiency. For a given sequence of caches, we define a *forward* schedule as follows. Assume we are at a cache, the distance to the immediately preceding cache is $p$ miles, and the total fuel at the cache, including the fuel in the jeep, is $f$ gallons. If $f - p \geq 1$, then go forward carrying a full gallon. Otherwise, go back, carrying $p$ gallons. If we are at the first cache, the start becomes the preceding cache. This schedule never carries back more fuel than the minimum needed to reach the preceding cache, and always goes forward starting with a full load of fuel.

With the forward schedule, whenever the jeep turns around, it goes all the way back to the start. We define a *hop* as moving between two adjacent caches, and a *trip* as beginning at the start, moving forward to some cache, then returning to the start.

For example, if we space the caches $1/3$ of a mile apart, a roundtrip to the first cache will burn $1/3$ of a gallon in each direction, and cache $1/3$ of a gallon. Two such trips will leave 2/3 of a gallon at the first cache. The third trip will burn 1/3 of a gallon getting to the first cache, pick up 1/3 of a gallon there, proceed to the second cache, deposit 1/3 of a gallon there, return to the first cache, burning another 1/3 of a gallon, pick up the 1/3 of a gallon remaining at the first cache, and return to the start.

Let $f(n)$ be the initial amount of fuel needed to deliver $1/3$ of a gallon to the $n$th cache, assuming all previous caches are empty. Then $f(1) = 1$, $f(2) = 2f(1) + 1 = 3$, $f(3) = 2f(2) + 2f(1) + 1 = 9$. In general, $f(n) = 1 + \sum_{i=1}^{n-1} 2f(i) = 3^{n-1}$. At three caches per mile, the cost of reaching $d$ miles and returning is $3^{3d-1} = O(27^d)$.

With caches $1/4$ of a mile apart, the cost of reaching $d$ miles is $O(16^d)$. This is much better than $O(27^d)$, but much worse than $O(7.389^d)$ for the optimal known-distance strategy. Since the exponential growth rate of these strategies is greater than for the optimal known-distance strategy, their competitive ratios are unbounded.

In fact, we will show that no strategy with a fixed sequence of caches can achieve a finite competitive ratio, in several steps. First, we will show that the forward schedule with any evenly spaced caches has an unbounded competitive ratio. Then we will show that the forward schedule is as efficient as any other schedule. Finally, we will show that for any number of caches, spacing them evenly gives the best performance.

**Forward schedule with evenly spaced caches.** The analysis of caches $1/3$ or $1/4$ mile apart is easy, because each trip empties all caches before the last cache visited. With more closely-spaced caches however, each roundtrip to a cache will leave fuel behind in some of the previous caches. This significantly complicates the analysis.

For example, consider caches $1/5$ mile apart. A trip to the first cache will burn $1/5$ of a gallon in each direction, and cache $3/5$ of a gallon. The second trip will arrive at the first cache bringing $4/5$ of a gallon, pick up $1/5$, travel to the second cache, leave $3/5$ of a gallon there, return to the first cache, pick up another $1/5$, and return to the start, leaving $1/5$ at the first cache. Reaching the second cache the next time is an easier problem than reaching it the first time, due to the residual fuel in the first cache.

The key to analyzing this algorithm is to divide the cost of a trip among the multiple goals that it serves. In the example above, the second trip contributes to reaching the second cache both the first and second times. While the actual algorithm always starts each trip with a full gallon, and hence the fuel used and number of trips is always an integer, we relax the original problem to allow splitting a trip into multiple trips that start with fractional fuel loads that sum to one. The continuous version of a problem is often much easier to analyze than the discrete version, and the optimal cost of the continuous relaxation of a problem is a lower bound on the cost of the discrete version, since the solution to the discrete version is also a solution to the continuous version.

**Theorem 1.** *For the forward schedule with evenly spaced caches* $1/k$ *miles apart, the asymptotic cost of reaching a given distance grows exponentially with the distance, and the base of the exponent is bounded from below by* $\left(\frac{k}{k-2}\right)^k$.

*Proof.* The first trip starts with a full gallon at the start, burns $1/k$ gallons each on the way to and from the first cache at $1/k$ miles, and leaves $1 - 2/k = \frac{k-2}{k}$ gallons at that cache. Thus, one gallon is needed to transfer $\frac{k-2}{k}$ gallons a distance of $1/k$ miles.

Since one gallon will transfer $\frac{k-2}{k}$ gallons $1/k$ miles, it takes $1/\frac{k-2}{k} = \frac{k}{k-2}$ gallons to transfer a full gallon to the first cache. To transfer a full gallon to the second cache, we need $\frac{k}{k-2}$ gallons at the first cache. Since it takes $\frac{k}{k-2}$ gallons to transfer one gallon to the first cache, it takes $(\frac{k}{k-2})^2$ gallons to transfer $\frac{k}{k-2}$ gallons to the first cache, and hence a full gallon to the second cache. Since there are $k$ caches per mile, transferring a full gallon one mile requires $(\frac{k}{k-2})^k$ gallons, and transferring a full gallon $d$ miles requires $(\frac{k}{k-2})^{kd}$ gallons.

For simplicity, we took as a milestone transferring a full gallon of fuel to a given cache, but the result is the same for reaching the very next cache, since that happens on the same trip, when using the forward schedule. Thus, the cost to initially reach each successive mile grows exponentially, and the base of the exponent is $(\frac{k}{k-2})^k$. This is only a lower bound on the actual cost, since we assumed fractional loads of fuel and trips, when only full loads and integer numbers of trips are allowed in practice. ∎

In fact, computer simulations of the forward schedule show that the asymptotic growth quickly converges to this exact value. The reason is that the residual fuel left behind with each trip is never wasted, but used on the next trip.

For $k = 5$, $\left(\frac{k}{k-2}\right)^k \approx 12.86$, for $k = 6$, $\left(\frac{k}{k-2}\right)^k \approx 11.39$, and for $k = 7$, $\left(\frac{k}{k-2}\right)^k \approx 10.54$. As $k$ goes to infinity, this quantity approaches $e^2 \approx 7.389$. Thus, for any finite value of $k$, the cost of this strategy grows exponentially faster than that of the optimal strategy, and thus the worst-case competitive ratio is unbounded.

There are two aspects of this strategy still to be addressed: the forward scheduling of hops, and the assumption of evenly spaced caches.

**The forward schedule is an optimal way to schedule hops.** It is never optimal to carry back more fuel than the minimum needed to reach the previous cache, and it is almost never optimal to start any hop with less than a full load. The only exception is the last hop to the first cache for the known-distance case if the actual distance doesn't equal one of the partial sums of the series. Within these constraints, the individual hops can be scheduled in different ways. For example, in the known-distance case, we could first transfer all the fuel needed between the start and the first cache, then transfer almost all this fuel to the second cache, etc., leaving behind only enough to get back to the start. The forward schedule is at the other end of the spectrum, always moving forward when there is enough fuel at a cache to carry a full load forward and still get back to the previous cache. There is also a range of intermediate schedules between these two. In fact, all schedules obeying the above two constraints have the same cost.

**Lemma 2.** *Forward scheduling of hops is optimal.*

*Proof.* Consider any three adjacent caches, $x$, $y$, and $z$, in order, with $x$ being closest to the start. In any solution to a given problem instance, a certain number of hops will be made from $x$ to $y$, and from $y$ to $z$. In an optimal solution, every such hop will start with a full load of fuel. The number of such hops is determined by the total amount of fuel needed at $z$. Regardless of the order in which these hops are made, the total number of hops from $x$ to $y$ and from $y$ to $z$ is the same, as long as there is always enough fuel at $y$ for every hop from $y$ to $z$ to carry a full load. Since the forward schedule obeys this constraint, it is an optimal schedule. ■

**Intermediate caches improve efficiency.** The efficiency of a fuel transfer is a property of one or more roundtrip hops. It is defined as the net amount of fuel added to the destination cache, times the distance from the source to the destination, divided by the amount of fuel removed from the source cache. The units are gallons transferred times miles, divided by total gallons. For simplicity, we use the fuel removed from the source, rather than the fuel burned. For a single hop, this is just the fuel added to the destination cache, times the distance traveled, since we always start with a full gallon. For example, a single hop between two caches $1/3$ of a mile apart adds $1/3$ of a gallon to the second cache, for an efficiency of $(1/3) \cdot (1/3) = 1/9$. Similarly, the efficiency of a single hop between two caches $1/4$ of a mile apart is $(1/2) \cdot (1/4) = 1/8$. This small increase in the efficiency of the individual hops translates into a large reduction in the cost of traveling $d$ miles, from $O(27^d)$ to $O(16^d)$. To measure the efficiency of a trip involving multiple hops, all the fuel at any intermediate caches must be removed.

**Lemma 3.** *For the forward schedule, given any pair of adjacent caches, we can always improve the efficiency of fuel transfers between them by adding an intermediate cache, with the most efficient placement being midway between the two caches.*

The intuition here is that as the amount of fuel on board decreases, the efficiency of transporting it decreases as well, since fuel mileage is constant. Adding an intermediate cache increases the average amount of fuel on board over the length of the trip.

*Proof.* We will compute the efficiency both with and without an intermediate cache. Since the total distance is the same in both cases, for comparison purposes we can define efficiency here as gallons delivered divided by gallons removed from the start.

Assume we have three caches $x$, $y$, and $z$ in a row, with $x$ closest to the start. Let the distance between $x$ and $y$ be $a$, and the distance between $y$ and $z$ be $b$. Without the cache at $y$, we go directly from $x$ to $z$ and back. Starting with one gallon at $x$, this delivers $1 - 2(a + b)$ gallons to $z$, for an efficiency of $1 - 2a - 2b$.

Now we add the intermediate cache at $y$. First, we make a roundtrip from $x$ to $y$, starting with one gallon, and delivering $1 - 2a$ gallons to $y$. Then we make a series of roundtrips from $x$ to $z$ and back to $x$, until all the fuel at $y$ is gone. Each such trip will take $a$ gallons from $y$ to replace the fuel burned from $x$ to $y$, and another $a$ gallons for the trip from $y$ back to $x$, for a net decrease of $2a$ gallons at $y$. Thus, $\frac{1-2a}{2a}$ such trips will exhaust the fuel at $y$. If $\frac{1-2a}{2a}$ is not an integer, we increase the number of gallons we start with at $x$ to $1 + m$, such that $m \cdot \frac{1-2a}{2a}$ is an integer, linearly scaling these quantities without affecting the efficiencies. Each such roundtrip will deliver $1 - 2b$ gallons of fuel to $z$, for a total of $\frac{(1-2a)(1-2b)}{2a}$ gallons. The total fuel started with is the number of such trips, plus the gallon for the initial round trip from $x$ to $y$, or $\frac{1-2a}{2a} + 1 = \frac{1}{2a}$. The efficiency is the fuel delivered divided by the fuel we started with, or $\frac{(1-2a)(1-2b)}{2a} \cdot 2a = 1 - 2a - 2b + 4ab$. Comparing the efficiency with and without the intermediate cache, we get $1 - 2a - 2b + 4ab$ vs. $1 - 2a - 2b$ or $4ab \geq 0$. Thus, the intermediate cache increases the efficiency. Furthermore, since the maximum product of two values that sum to a constant occurs with equal values, the maximum efficiency occurs when $a = b$ and $y$ is midway between $x$ and $z$. ∎

Why doesn't this result also apply to the known-distance case? In that case, a fixed amount of fuel is transferred between each pair of adjacent caches, and the distances between the caches are chosen so that every hop starts with a full load. Adding an intermediate cache will improve the efficiency of most of the hops, but the last hop from the intermediate cache will start without a full load, negating the previous gains.

**Lemma 4.** *For the unknown-distance case, in an optimal solution for a fixed sequence of caches, the caches must be evenly spaced.*

*Proof.* Assume that we have an optimal solution where the caches are not evenly spaced. Then there must be three adjacent caches $x$, $y$, to $z$, where the distance from $x$ to $y$ is not equal to the distance from $y$ to $z$. Lemma 3 tells us that we can improve the efficiency of transfers in this segment by repositioning $y$ equally distant between $x$ and $z$, thus contradicting our assumption that the existing solution is optimal. ∎

Now we can state and prove our general result for fixed caches.

**Theorem 5.** *No strategy for the unknown-distance case with a fixed sequence of caches can achieve a finite worst-case competitive ratio.*

*Proof.* Assume we have an optimal strategy for the unknown-distance case with a fixed sequence of caches. Lemma 2 tells us that we can schedule the individual hops using the forward schedule, with no loss of efficiency. Lemma 4 tells us that the caches must be evenly spaced. Finally, Theorem 1 tells us that the cost of the forward schedule with caches $1/k$ miles apart grows exponentially with distance, and that the base of the exponent is at least $\left(\frac{k}{k-2}\right)^k$. Since the base of the exponent for the optimal strategy for a known distance is $e^2$, which is less than this value for any finite $k$, the competitive ratio of any such strategy is infinite. ∎

The only way to obtain the $e^2$ rate of growth for the unknown-distance case with a fixed sequence of caches is for the interval between adjacent caches to go to zero, resulting in an infinite number of caches in any finite interval. Since the forward schedule requires more than a full load of fuel at any cache to progress beyond it, this would require caching an infinite amount of fuel to move any distance beyond the start.

I found this result very surprising. When I began this work, I was convinced that there was an optimal strategy with a fixed sequence of caches, and set out to find it.

## 6. ITERATIVE STRATEGIES.

If the sequence of caches is not fixed, then it must change at some time. Adding or removing caches beyond the furthest point reached at that time has no effect, and is the same as having those caches from the beginning. If we add new caches in a region that has already been traversed, then Lemma 3 tells us that it's more efficient to include the new caches from the beginning. Similarly, removing caches in a region that has already been covered decreases efficiency.

The only other option is to remove the fuel from all caches, and start over with a new sequence of caches, since it would be wasteful to leave any fuel behind. Whenever we change caches, we will have advanced some distance from the start, and the only thing we will have learned is that the goal is further away. Thus, we can determine these points in advance, and whatever strategy was used up to that point can be replaced with a strategy that uses the optimal sequence of caches for that known distance.

Thus, we are left with a strategy that chooses a sequence of target distances, and uses the known optimal strategy for each distance, emptying the caches in between. We refer to this as an iterative strategy. The intuition behind this is that since the cost of reaching a given distance via the optimal strategy grows exponentially with distance, the cost of the unsuccessful iterations will not affect the asymptotic cost of the algorithm, giving rise to a finite competitive ratio. The remaining challenge is to find a sequence of target distances that minimizes the worst-case competitive ratio.

**Computing the worst-case competitive ratio.** The goal is always found on the last iteration of an iterative strategy, since the algorithm terminates once it finds the goal. Thus, the range of the last iteration must equal or exceed the distance to the goal, and the range of the previous iteration must be less than the distance to the goal. We use the forward schedule in order to reach the goal as soon as possible.

In practice, where the goal is found in the last iteration will have little effect on the cost of that iteration. For simplicity, we measure the cost of an iteration by the amount of fuel taken from the start, or equivalently the total distance traveled if we reach the target distance of the iteration. Regardless of where the goal is found on the last iteration, this will be an upper bound on the distance traveled, giving us an upper bound on the competitive ratio. Thus, we use the full cost of the last iteration as the cost of that iteration, regardless of where in that iteration the goal is found.

Where the goal is found in the last iteration of an iterative strategy does makes a significant difference in the optimal cost of finding it had we known its distance in advance, however, since the cost of such an optimal solution is $O(e^{2d})$.

Recall that the competitive ratio of a strategy for the unknown-distance case is the actual cost to find the goal divided by the optimal cost to find it, had we known its distance in advance. We refer to "the competitive ratio of an iteration" as the competitive ratio if the goal is found in that iteration. The actual cost of finding the goal in an iteration is the sum of the costs of all iterations up to and including that iteration. Since the goal must be found past the range of the penultimate iteration, the optimal cost is minimized, and hence the competitive ratio of the iteration is maximized, if the goal is found just past the range of the penultimate iteration, and hence we use the cost of that iteration as the optimal solution cost. This also gives us an upper bound on the competitive ratio. Thus the worst-case competitive ratio of an iteration is the sum of the cost of all iterations up to and including that iteration, divided by the cost of the penultimate iteration. The worst-case competitive ratio is the maximum competitive ratio of all iterations in the infinite sequence of iterations.

What sequence of target distances will minimize the worst-case competitive ratio? Our analysis is considerably simplified by representing the iteration costs not by their actual costs, but by a sequence of multipliers or ratios of the costs of two consecutive iterations, as follows: Let $m_1$ be the cost of the first iteration. Let $m_2$ be the cost of the second iteration divided by the cost of the first iteration, making the cost of the second iteration $m_1 m_2$. Similarly the cost of the third iteration is $m_1 m_2 m_3$, etc.

The competitive ratio of the $n$th iteration is the sum of the cost of all $n$ iterations, divided by the cost of the $(n-1)$st iteration, or

$$\frac{m_1 + m_1 m_2 + m_1 m_2 m_3 + \cdots + m_1 m_2 \cdots m_{n-1} + m_1 m_2 \cdots m_n}{m_1 m_2 \cdots m_{n-1}}$$
$$= \frac{1}{m_2 m_3 \cdots m_{n-1}} + \frac{1}{m_3 m_4 \cdots m_{n-1}} + \cdots + \frac{1}{m_{n-2} m_{n-1}} + \frac{1}{m_{n-1}} + 1 + m_n.$$

We discuss how to compute the competitive ratio of the first iteration at the end of this section under the heading "Optimal initial iterations."

**Lemma 6.** *There exists an optimal iterative strategy where the competitive ratios associated with finding the goal in any iteration are all equal.*

*Proof.* Given a sequence of iterations, we will transform it into one with equal competitive ratios, without increasing the worst-case ratio. If the competitive ratio of the first iteration is less than the worst-case competitive ratio of the whole sequence, then increase $m_1$ until its competitive ratio equals this value. The remaining multipliers are unchanged, and the effect of this is to increase the costs of all subsequent iterations, but not their competitive ratios, which are all independent of $m_1$. Then for each iteration $n$ whose competitive ratio is less than that of the first iteration, increase $m_n$ until its competitive ratio equals that of the first iteration. Increasing $m_n$ will not affect the competitive ratio of any previous iteration, will increase the competitive ratio of the $n$th iteration, and will decrease the competitive ratio of each subsequent iteration, since $m_n$ appears only in the denominators of the competitive ratios of those iterations. By repeating this process for every iteration, we get a new strategy with all competitive ratios equal to the maximum competitive ratio of the original strategy. ■

Given a strategy with unequal competitive ratios, we can actually reduce their maximum by setting them all equal, but we do not need this stronger result. We assumed above that the iteration costs are real numbers, when in fact they are integers, since every trip starts with a full gallon of fuel. Since the fuel costs grow exponentially with distance, the relative difference between the real values and the nearest integers goes to zero as the iteration distances increase.

**Lemma 7.** *In any iterative strategy in which the competitive ratios are all equal, the multipliers starting with $m_2$ form a decreasing sequence.*

*Proof.* The proof is by induction on the length of the sequence.

Base case: First we show that $m_3 < m_2$. The competitive ratio of the second iteration is $\frac{m_1+m_1m_2}{m_1} = 1 + m_2$, and the competitive ratio of the third iteration is $\frac{m_1+m_1m_2+m_1m_2m_3}{m_1m_2} = \frac{1}{m_2} + 1 + m_3$. Settings these ratios equal, we get

$$1 + m_2 = \frac{1}{m_2} + 1 + m_3 \implies m_2 = \frac{1}{m_2} + m_3 \implies m_3 < m_2.$$

Induction step: Assume that for all $m$ between $m_2$ and $m_n$ inclusive, $m_i < m_{i-1}$. We want to prove that $m_{n+1} < m_n$. The competitive ratio of the $n$th iteration is

$$\frac{1}{m_2 m_3 \cdots m_{n-1}} + \frac{1}{m_3 m_4 \cdots m_{n-1}} + \cdots + \frac{1}{m_{n-2} m_{n-1}} + \frac{1}{m_{n-1}} + 1 + m_n.$$

Similarly, the competitive ratio of the $(n+1)$st iteration is

$$\frac{1}{m_2 m_3 \cdots m_n} + \frac{1}{m_3 m_4 \cdots m_n} + \cdots + \frac{1}{m_{n-1} m_n} + \frac{1}{m_n} + 1 + m_{n+1}.$$

Setting these values equal to each other, and subtracting 1 from both sides, we get

$$\begin{aligned}&\frac{1}{m_2 m_3 \cdots m_{n-1}} + \frac{1}{m_3 m_4 \cdots m_{n-1}} + \cdots + \frac{1}{m_{n-2} m_{n-1}} + \frac{1}{m_{n-1}} + m_n \\ &\quad = \frac{1}{m_2 m_3 \cdots m_n} + \frac{1}{m_3 m_4 \cdots m_n} + \cdots + \frac{1}{m_{n-1} m_n} + \frac{1}{m_n} + m_{n+1}.\end{aligned}$$

By the induction hypothesis, we know that $m_n < m_{n-1}$, and hence $\frac{1}{m_n} > \frac{1}{m_{n-1}}$, and we can rewrite $\frac{1}{m_n}$ as $\frac{1}{m_{n-1}} + x_{n-1}$, for some positive $x_{n-1}$. Similarly, since $m_{n-1} < m_{n-2}$ and $m_n < m_{n-1}$, we have $m_{n-1}m_n < m_{n-2}m_{n-1}$ and $\frac{1}{m_{n-1}m_n} > \frac{1}{m_{n-2}m_{n-1}}$, and we can rewrite $\frac{1}{m_{n-1}m_n}$ as $\frac{1}{m_{n-2}m_{n-1}} + x_{n-2}$ for some positive $x_{n-2}$. In general, we can rewrite $\frac{1}{m_i m_{i+1} \cdots m_n}$ as $\frac{1}{m_{i-1} m_i \cdots m_{n-1}} + x_{i-1}$ for some positive $x_{i-1}$. Thus, we can rewrite the above equation as

$$\begin{aligned}&\frac{1}{m_2 m_3 \cdots m_{n-1}} + \frac{1}{m_3 m_4 \cdots m_{n-1}} + \cdots + \frac{1}{m_{n-1}} + m_n \\ &\quad = \frac{1}{m_2 m_3 \cdots m_n} + \frac{1}{m_2 m_3 \cdots m_{n-1}} + x_2 + \cdots + \frac{1}{m_{n-1}} + x_{n-1} + m_{n+1}.\end{aligned}$$

Subtracting common terms from both sides of the equation gives us

$$m_n = \frac{1}{m_2 m_3 \cdots m_n} + x_2 + x_3 + \cdots + x_{n-1} + m_{n+1} \implies m_{n+1} < m_n$$

which is what we were trying to prove. Thus, by induction, for all $i \geq 2$ we have $m_{i+1} < m_i$, and the sequence of multipliers is a decreasing sequence. ∎

Finally, we can state and prove the main result of our article.

**Theorem 8.** *One optimal strategy for the unknown-distance case, as measured by the worst-case competitive ratio, is an iterative strategy that doubles the cost of each successive iteration, and achieves a worst-case competitive ratio of four.*

*Proof.* We showed that the only optimal alternative to a fixed sequence of caches is an iterative strategy that uses the optimal known-distance strategy to reach each of a sequence of successively greater target distances. Lemma 6 tells us that given such a strategy, there is an equally efficient one in which all the competitive ratios are equal. Lemma 7 tells us that in such a strategy, the multipliers of each successive iteration form a decreasing sequence. The multipliers must all be greater than one, since each iteration must go further than the previous one. Thus, they must converge to a limit greater than or equal to one. The competitive ratio for a given iteration can be written as a sum of terms, including the multiplier for that iteration, one, and a sequence of terms, each of which is the reciprocal of a product of an increasingly longer sequence of multipliers. There are three cases to consider: (1) the multipliers converge to one, and their products converge to a finite value; (2) the multipliers converge to one, and their products are unbounded; and (3) the multipliers converge to a value greater than one, and thus their products must be unbounded. In case 1, the reciprocals of the products converge to a nonzero value. The competitive ratios are the sums of these reciprocals, and since there can be an infinite number of iterations, the worst-case competitive ratio will be unbounded. This is not optimal, since we can achieve a finite competitive ratio. In either case 2 or 3, the products grow without bound, and their reciprocals go to zero. Thus the terms including the early multipliers go to zero, and the competitive ratio is determined by the limiting value $m$ of the multipliers.

What value of $m$ gives the lowest competitive ratio? The competitive ratio is the infinite sum $m + 1 + \frac{1}{m} + \frac{1}{m^2} + \frac{1}{m^3} + \cdots$. Since $\sum_{i=0}^{\infty} m^{-i} = \frac{m}{m-1}$, the competitive ratio is $m + \frac{m}{m-1} = \frac{m^2}{m-1}$. The first derivative of this expression is $\frac{m^2-2m}{(m-1)^2}$. Setting this equal to zero gives us $m = 2$, and a worst-case competitive ratio of 4. ■

Thus, each successive iteration should double the cost of the previous iteration. How much additional distance $x$ does that correspond to? Since reaching a distance $d$ costs $O(e^{2d})$, reaching $d + x$ costs $O(e^{2(d+x)})$. Setting this equal to twice the cost of the previous iteration gives us the equation $2e^{2d} = e^{2(d+x)}$. Solving this equation for $x$ gives us $x = (\ln 2)/2 \approx .3465736$. Thus, one optimal solution to the unknown-distance case is a series of iterations, each a little over a third of a mile longer than the previous one, clearing all caches between iterations.

**Optimal initial iterations.** We can actually improve on the simple doubling strategy, and still achieve a worst-case competitive ratio of 4. To compute the competitive ratios here, we assess the cost of an iteration as the total distance traveled to the goal and back, or equivalently the amount of fuel burned, as opposed to the amount of fuel removed from the start. The longest first iteration with a competitive ratio of 4 or less uses 8 gallons to go 1.359 miles. The worst-case ratio occurs when the goal is found just past the fifth cache, .44226 miles from the start. The second iteration starts with 25 gallons to go 1.9 miles. Since this is more than 1/2 mile further than the previous iteration, this iteration never reaches the last cache, saving a little more than a gallon of fuel. Since $(8 + 24)/8 = 4$, this iteration has a competitive ratio less than 4.

For each subsequent iteration, in the worst case the goal is found just past the last cache, saving less than a gallon. For example, the total cost of the first three iterations is almost $8 + 25 + 67 = 100$, and the goal is found just past the range of the second iteration, for a competitive ratio just under $(100/25) = 4$. Table 1 shows the first ten such optimal iterations, with the fuel used, distance traveled, additional distance beyond the previous iteration, competitive ratio, and fuel ratio to that of the previous iteration. As we extend the sequence, the fuel ratios approach 2, and the competitive ratios approach 4. This saves multiple iterations over the pure doubling strategy, starting from 1. For example, the 7th iteration in this sequence goes further than the 12th iteration in the doubling strategy, and the number of iterations saved continues to grow.

## 7. CONCLUSIONS.

We considered the classic problem of a jeep crossing a desert that is wider than its fuel range, but where we do not know the distance in advance. We evaluated the efficiency of a strategy by its worst-case competitive ratio, which is the largest ratio of the actual cost to find the goal, divided by the optimal cost had we known the distance in advance. Surprisingly, we found that any strategy based on a fixed sequence of caches cannot achieve any finite competitive ratio. Rather, the

**Table 1.** First 10 Iterations in the optimal solution.

| Iter. | Fuel | Distance | Add. Dist. | Comp. Ratio | Fuel Ratio |
|---|---|---|---|---|---|
| 1 | 8 | 1.358929 | | 3.652759 | |
| 2 | 25 | 1.907979 | .549051 | 3.987737 | 3.125000 |
| 3 | 67 | 2.394676 | .486697 | 3.961064 | 2.680000 |
| 4 | 168 | 2.852076 | .457400 | 3.986346 | 2.507463 |
| 5 | 404 | 3.289934 | .437858 | 3.994787 | 2.404762 |
| 6 | 944 | 3.713936 | .424002 | 3.997901 | 2.336634 |
| 7 | 2160 | 4.127655 | .413719 | 3.999123 | 2.288136 |
| 8 | 4864 | 4.533467 | .405812 | 3.999624 | 2.251852 |
| 9 | 10816 | 4.933022 | .399555 | 3.999836 | 2.223684 |
| 10 | 23808 | 5.327507 | .394485 | 3.999927 | 2.201183 |

optimal solution is an iterative one where we use the optimal known-distance strategy to reach each of a sequence of greater target distances. One optimal iterative strategy doubles the cost of each successive iteration, and achieves a worst-case competitive ratio of four. In fact, we can more than double the cost of each iteration, but the ratio of the costs of successive iterations approaches two in the limit of large distance.

**ACKNOWLEDGMENTS.** I would like to thank Michael Christ and James Ralston for helpful discussions concerning this problem. I would also like to thank the two anonymous referees, whose comments greatly improved the paper.

REFERENCES

[1] Bailey, H. (2009). Jeeps penetrating a hostile desert. *Coll. Math. J.* 40(3): 182–187.
[2] Dewdney, A. K. (1987). Computer recreations. *Sci. Am.* 256(6): 128–131.
[3] Fine, N. J. (1947). The jeep problem. *Amer. Math. Monthly.* 54(1): 24–31.
[4] Franklin, J. N. (1960). The range of a fleet of aircraft. *J. Soc. Ind. Appl. Math.* 8(3): 541–548.
[5] Gale,D. (1970). The jeep once more, or jeeper by the dozen. *Amer. Math. Monthly.* 77(5): 493–501.
[6] Haurath, A., Jackson, B., Mitchem, J., Schmeichel, E. (1995). Gale's rountrip jeep problem. *Amer. Math. Monthly.* 102(4): 299–309.
[7] Phipps, C. G. (1947). The jeep problem: a more general solution. *Amer. Math. Monthly.* 54(8): 458–462.
[8] Sleator, D. D., Tarjan, R. E. (1985). Amortized efficiency of list update and paging rules. *Comm. ACM.* 28(2): 202–208.

**RICHARD KORF** earned his Ph.D. in computer science from Carnegie-Mellon University in 1983. He is a Professor of Computer Science at UCLA. He usually works on heuristic search in artificial intelligence. In his spare time he enjoys mountaineering.
*Department of Computer Science, University of California, Los Angeles, Los Angeles, CA 90095*
*korf@cs.ucla.edu*