

The logo for ClickSoftware, featuring the company name in a bold, blue, sans-serif font. The 'i' in 'Click' has a yellow dot above it, and the 'S' in 'Software' has a yellow dot to its right. The background of the logo is a white circle with a yellow gradient.

Making Service Click

A detailed map of the Birmingham area in the UK, showing streets, landmarks, and major roads like the M42 and M40. The map is overlaid with a semi-transparent white car and a large yellow gear. The car is a white hatchback with a black stripe along the side, and the gear is a large, stylized yellow gear with a white center. The text 'Street-Level Routing: Where the Rubber Meets the Road' is overlaid on the map and car.

# Street-Level Routing: Where the Rubber Meets the Road

**A ClickSoftware White Paper**

**Copyright Notice**

Copyright © 2006 ClickSoftware Technologies Ltd. All rights reserved.

No part of this publication may be copied without the express written permission of: ClickSoftware Technologies Ltd.

**Publication Notice**

The information contained herein does not constitute a warranty of any kind. ClickSoftware Technologies Ltd. reserves the right to revise this publication and make changes without notification. ClickSoftware Technologies Ltd. assumes no liability for losses incurred as a result of information contained herein.

## Table of Contents

1	Introduction: People drive on roads, not in the air	1
2	Some examples	2
2.1	Example 1: Who is closest?	3
2.2	Example 2: Where to go first?	4
2.3	Example 3: Along the way	5
2.4	Example 4: The high road or the low road?	7
3	Things that go bump in the Gantt: the impact of wrong travel-time estimates	9
4	Alternatives to true SLR	11
4.1	Showing the map	12
4.2	Using postal codes	12
4.3	Optimize one route at a time	14
4.4	Precalculated travel between all known service locations	15
5	Yes, but...	16
6	Conclusion	16

## 1 Introduction: People drive on roads, not in the air

*"Bypasses are devices that allow some people to dash from point A to point B very fast while other people dash from point B to point A very fast. People living at point C, being a point directly in between, are often given to wonder what's so great about point A that so many people from point B are so keen to get there, and what's so great about point B that so many people from point A are so keen to get there. They often wish that people would just once and for all work out where they wanted to be."*

**Douglas Adams, *The Hitchhiker's Guide to the Galaxy***

Indeed, as Douglas Adams pointed out, most people do not feel it is particularly useful to just drive around. When they need to be in several different places during the day, they try to plan their route in the most efficient manner, so that their travel time between these places becomes as short as possible. To do so, they look for bypasses, bridges, and any other means of shortening their travel – because unlike birds or airplanes, cars often can't take the straight-line route between two points.

The previous point is so self-evident that you might wonder why a whole white paper is dedicated to explaining the benefits of knowing the true drive time between two points, given the roads connecting these points, rather than assuming that this drive time is directly related to the straight-line distance. Well, as you read this paper you may find some surprises and not-so-obvious implications of the right and wrong ways of planning your organization's routes. Also, not everybody uses street-level routing (SLR) in planning schedules and routes for mobile work forces. In fact, some would go out of their way to explain why the intuitively correct way – using SLR – is not the right thing to do. It is interesting to examine the arguments made in defense of such claims.

The objective of this white paper, then, is to show that SLR creates much better routes and schedules than any other method. Along the way, we will look at the impact of badly estimated travel times on operational efficiencies, customer satisfaction and productivity; examine some alternatives to SLR; explain why many vendors do not offer SLR; and present some quantitative, real-life evidence for the importance of SLR.

The following themes are threaded through much of this paper:

- **Optimize across the workforce, not one route at a time:** Street-level routing is not mainly about how a specific mobile worker should drive from point A to point B; its key productivity improvement is about deciding which mobile should be dispatched to point B.
- **Avoid pitfalls:** Street-level routing discovers hidden traps in planning and defuses them before they create delays, missed commitments, customer dissatisfaction and reduced productivity.
- **The map is your friend:** Street-level routing not only exposes constraints on productivity but also discovers unexpected productivity-improvement gains.

These messages should become clearer as you read through the examples and explanations.

## 2 Some examples

*Alice: Would you tell me, please, which way I ought to go from here?*

*The Cheshire Cat: That depends a good deal on where you want to get to.*

**Lewis Carroll, *Alice in Wonderland***

This section presents several examples of how identical tasks would be scheduled and dispatched by a linear-distance-based solution compared to an SLR-based system. Some of these examples emphasize the lesson that the Cheshire Cat taught Alice: In the mobile service world, the important decision is which tasks should be assigned to which mobile workers. If you don't use all the available information while making that decision, your productivity could be significantly harmed.

Note: For clarity, these examples ignore many important parts of the scheduling challenge, such as the promised time window – the time during which the organization committed to arrive at the service site; the worker-task match – the match of the required task to the assigned mobile worker's skills, certification, or customer relationship; the availability of goods, spare parts and tools required to perform the task; and much more. In most cases, the inclusion of these factors in mobile workforce scheduling makes the importance of SLR even more pronounced.

## 2.1 Example 1: Who is closer?

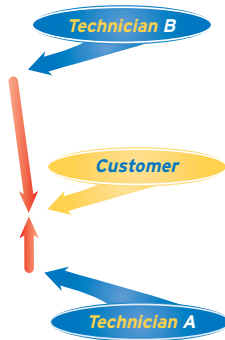


Figure 1: Linear distances: Which technician is closer?

Consider Figure 1. All else being equal, which technician should we dispatch to the customer? It seems clear that Technician A is closer.

However, Figure 1 does not tell the whole story. Figure 2 shows the same locations on a map. Technician A would have to drive a long way to the northwest in order to get on M5 going northeast, then cross the bridge and drive southeast to the point that seemed – in Figure 1 – to be just a short drive to the north.

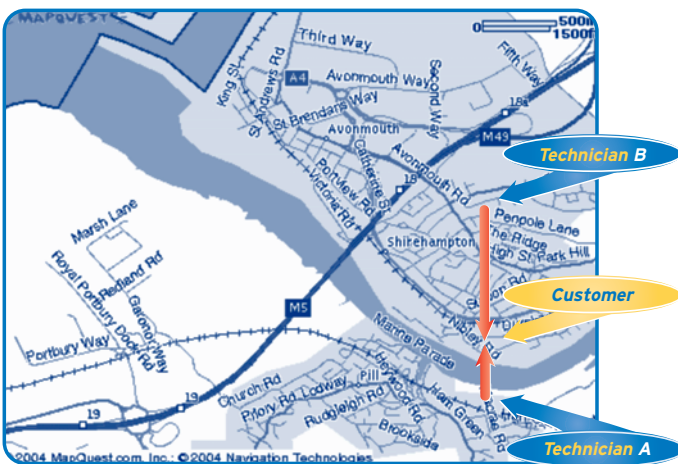


Figure 2: Street-Level Routing: Which technician is closer?

Granted, not all points have a river flowing between them. Yet there are a good number of similar geographic “obstacles,” such as slow-traffic urban areas, or having to cross a fast motorway by

traveling to the nearest overpass. This implies that most schedules created without considering SLR would have quite a few such significant obstacles. Each such obstacle translates into missed appointments for the customer waiting beyond that obstacle – and probably for all other customers scheduled on this technician’s route for the remainder of the day.

## 2.2 Example 2: Where to go first?

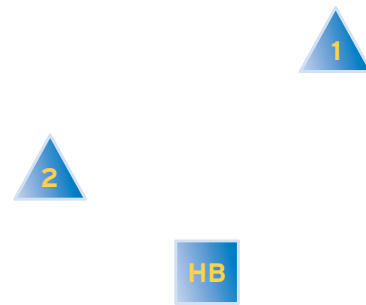


Figure 3: Linear distances: Where to go first?

In Figure 3, “HB” stands for the Home Base of one mobile worker. We want this worker to handle tasks 1 and 2. Looking only at travel distance<sup>1</sup>, the path HB → 2 → 1 is shorter than the path HB → 1 → 2. Therefore, we should dispatch task 2 first.

As you might expect, the real travel paths tell a different story: In Figure 4, we see that the path HB → 1 → 2 is significantly shorter than the path HB → 2 → 1.



Figure 4: Street-Level Routing: Where to go first?

<sup>1</sup> In this example we’re ignoring the need to return to the Home Base, since there will be other tasks scheduled after tasks 1 and 2 are completed.

### 2.3 Example 3: Along the way

This example is a simple variation of the previous example. This time, we have two technicians, imaginatively named Technician 1 and Technician 2, starting the day at Home Bases HB1 and HB2 respectively.

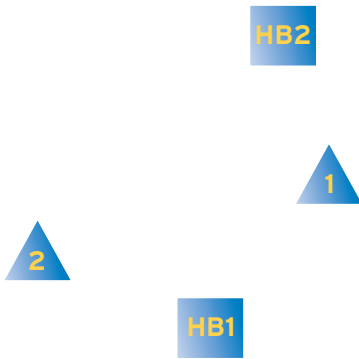


Figure 5: Linear Distance: How to assign tasks 1 and 2?

In Figure 5, it seems reasonable to assign task 2 to Technician 1 and task 1 to Technician 2. This way, the load is shared across the available workforce and the total travel distance is low.

Figure 6 shows the routes both technicians would need to travel to reach the location of the required tasks. Now it becomes clear that if Technician 1 is assigned to task 2, it costs us almost no extra travel to assign task 1 to the same Technician. The solution that previously seemed attractive is now revealed to require substantially more travel. In this case, we're better off assigning Technician 1 to both tasks, and finding other tasks for Technician 2 to perform.<sup>2</sup>

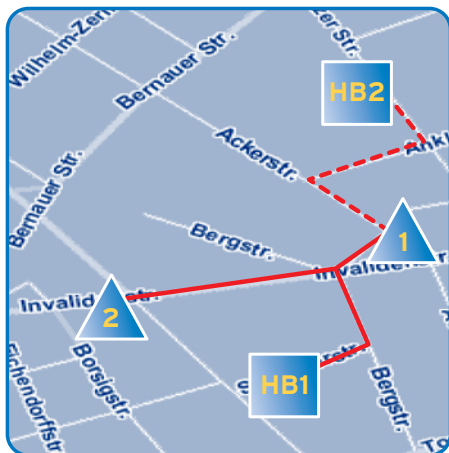


Figure 6: Street-Level Routing: How to assign tasks 1 and 2?

<sup>2</sup> In this example, as in the other examples, the assumption is that the shown tasks and mobile workers are just a part of the overall tasks and resources that need to be scheduled.

### 2.4 Example 4: The high road or the low road?

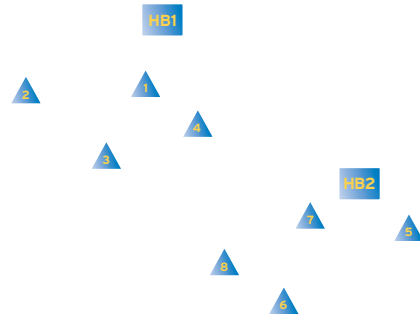


Figure 7: Linear Distance: Which technician should take which jobs?

In Figure 7, our protagonists, Technician 1 and Technician 2, face a larger challenge. Now they need to divide eight tasks between them. Blissfully unaware of the street map, they decide to cluster the tasks by their locations. This gives tasks 1, 2, 3 and 4 to Technician 1, with the rest assigned to Technician 2.

As they start their engines and the rubber meets the road, Figure 8 shows what they discover: the paths between the tasks assigned to each of them flow through small one-way streets, necessitating slow and meandering routes.



Figure 8: Street-Level Routing: The linear-distance-based solution meets the actual streets

Could there be a better assignment? Figure 9 shows a somewhat counterintuitive routing based on maximizing the use of faster roads (marked in yellow on this map) and avoiding twists and

turns through the slow and narrow streets. Today's route optimization offerings are well aware of the fact that not all streets and roads allow movement at the same speeds, and the routes in Figure 9 make full use of this capability. Note that even though Technician 1 starts the day driving "the long way around" to task 2 instead of greedily grabbing the nearby task 1, this "long way around" takes a fast road and positions Technician 1 where tasks 3, 8 and 6 can be picked up one after the other, with small detours from road 140, while Technician 2 does the same along road 129.<sup>3</sup>



**Figure 9: Street-Level Routing:**  
The SLR-based solution maximizes the use of faster roads

### 3 Things that go bump in the Gantt: the impact of wrong travel-time estimates

...ghoulies and ghosties  
And long-leggedy beasties  
And things that go bump in the night...  
**Scottish Saying**

When our estimate of the travel distance between points A and B is wrong, we pay more for fuel, vehicle wear and tear, periodic maintenance, etc.<sup>4</sup> What happens when our estimate of the drive time between A and B is wrong? The impact extends beyond increased operational costs for the simple reason that travel time estimates drive estimated arrival times.

Suppose that a mobile worker has six tasks to handle today and that we've managed to estimate travel time for almost all the

route segments correctly, except for the segment between the third and the fourth tasks. If we're lucky, we overestimated the travel time and merely lost the chance to raise productivity by squeezing in yet one more task. If we underestimated the travel time, we get an avalanche of problems, starting from being late to reach the location of the fourth task – possibly breaking the commitment made to that customer. How about the fifth and sixth task? The dispatcher will now start looking for a way to regain lost time, maybe by finding an available mobile worker who is close enough to take the fifth task. However, efficient organizations try to match work capacity to the demand for service: this late in the day, most workers are committed to other tasks. Therefore, if a solution exists, it probably involves a reshuffling involving several employees, or the postponement of a low-urgency task to the next day. These decisions, whether made by optimization software or by human dispatcher, face so many constraints that they can't yield a solution as good as the one we could have planned in the morning, if only we'd had access to better estimates.

To sum up, an unexpected "bump" that delays a worker's arrival to just one task but bumps all the following tasks for the same worker risks the following repercussions:

- Failure to meet the terms of the SLA (service level agreement) – leading to decreased customer satisfaction and sometimes creating legal liability
- Paying overtime as a result of bumping the schedule beyond standard working hours
- Having to dispatch higher-cost workers – for example, subcontractors
- "Fire-fighting" reassignments in the middle of the day: intensive manual work, which has few opportunities to succeed since some of the mistakes are impossible to fix at the time the problem is detected
- Losing employee confidence in the scheduling process
- Missing KPI (key performance indicator) targets, which may lead to issues with regulatory governance organizations
- Losing customers

<sup>3</sup> Interestingly, the technicians in Figure 9 end their routes near each other's home bases. However, this is not a problem since they can use the same fast roads to get back to their own home base.

<sup>4</sup> The error in using linear distance ("as the crow flies") is almost always an underestimate, unless we artificially multiply all linear distances by some factor – a method that has the dubious advantage of being capable of both overestimates and underestimates.

Good travel time estimates, on the other hand, not only drastically reduce this kind of impact, they also lead to better schedules and higher productivity, as shown by some of the above examples. Specifically, examples 3 and 4 show how a careful analysis of the map can lead to squeezing more tasks into a technician's day, compared to what we would expect without having access to the map. In other words, relying on "as the crow flies" distance and travel-time estimates loses out from both sides: the performance it expects<sup>5</sup> is lower than the actual performance it leads to and expected performance is lower than the performance achievable with true SLR-based routing. Figure 10 is a visual representation of the impact of scheduling based on linear-distance versus scheduling based on SLR.

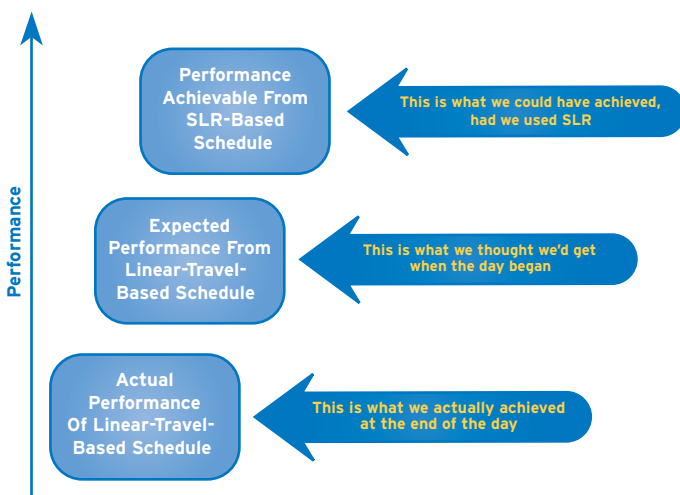


Figure 10: Performance impact of scheduling based on linear-distance vs. SLR

## 4 Alternatives to true SLR

If true SLR is so good, why aren't all schedule-optimization companies using it? The answer is the computational difficulty. To understand this, think of a work group having 120 mobile workers who need to handle about 800 tasks per day. To use SLR to create the schedule, the real travel time between each pair of tasks needs to be calculated: For each of the 800 points, we need to calculate the route to each of the other 799 points – the route and drive time from point A to point B is usually not the same as

for the opposite direction. This requires 800\*799 calculations – over 600,000 different point-to-point routes.<sup>6</sup> Such calculations require a lot of computing time, since the computer needs to consider many combinations of road segments before finding the best one. We typically expect no more than 20 calculations per second from a reasonably powerful server. Therefore, the overall calculation of just these pair-to-pair distances would take about 30,000 seconds, or over eight hours. Multiplying the organization size by only two would multiply the computation time by four – over 30 hours as preparation for optimizing tomorrow's schedule. At this rate, even sharing the computation load across more or faster servers cannot keep up with the rapid growth in required computation time.

The above is only one difficulty on the road to true SLR. So is true SLR impossible? The answer is no. ClickSoftware, through extensive R&D effort and through refinement in many field-proven deployments, has shown that it is indeed possible.

Another question that may be asked is whether true SLR is really needed: Aren't there ways to avoid the pitfalls of straight-line-based scheduling without requiring SLR? We now turn to a review of some approaches that strive to avoid the drawbacks of linear-distance-based scheduling without reaching the level of true SLR.

### 4.1 Showing the map

This option isn't really an alternative to SLR, though it sometimes seems to be presented as such. Some scheduling software ignores the street-based calculation of travel times and distances, but it does display the locations of tasks, and of mobile workers, on a map of the area. Furthermore, when asked, the software will give travel directions, with estimated drive time, for any route between two points. These directions would be made available to the managers and dispatchers, and – if the right mobile infrastructure were available – on mobile devices as well. Since these directions are accessible, and are based on the street map, it is easy to forget that the software itself does not use these travel time estimates. At best, it might allow humans to "fix" the schedule by reviewing the routes displayed on the map, detect discrepancies between expected arrival time and the actual route, and tweak the schedule accordingly – a manual, work-intensive and error-prone process.

<sup>5</sup> In this context, performance may be measured by meeting SLA terms, by costs (e.g., travel and overtime) and in utilization (e.g., number of tasks completed).

<sup>6</sup> Sometimes reductions are possible, as when it is clear that some pair of tasks would never be performed by the same mobile worker (e.g., because the tasks have incompatible skill requirements). Other factors also affect the number of calculations, such as the need to compute travel times from and to the mobile worker's home base, but these reductions and factors do not materially affect the magnitude of this large computational task.

## 4.2 Using postal codes

The rationale for this method is the following: If we could group the tasks into meaningful groups by their locations, we would need to calculate only the intergroup routes. This leads to a sharp decline in required computation: If each group only has 10 tasks, the number of intergroup routes is 100 times less than the number of intertask routes.

The simplest way to group tasks is by the postal code (ZIP code in the USA) of the address where the task needs to be performed. Postal codes are numbers used by the postal service to identify an area where mail is delivered. This has several advantages apart from the obvious one – ease of classifying tasks into groups. One such advantage is that the postal service attempts to avoid including both sides of a river, for example, in the same postal code<sup>7</sup>. Another is that since postal-code areas rarely change, interarea calculations may be done once and their results stored for quick access.

In this method, a postal code area is treated as if all the tasks within it are located at the area's "centroid." The centroid is a point within the postal code area defined either geometrically or based on population density – see Figure 11.

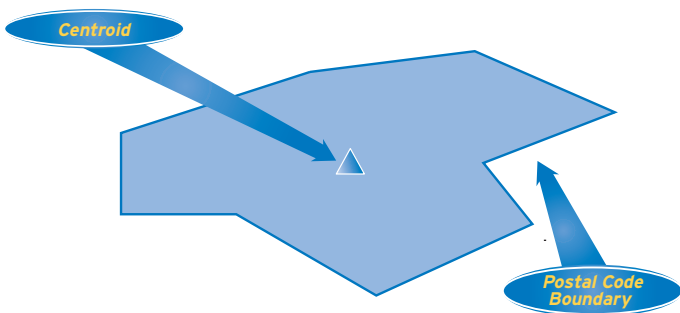


Figure 11: Postal-code area with centroid

Figure 12 clarifies the treatment of locations within postal-code areas: Points A and B will be considered to be at the exact same location (but not point C).

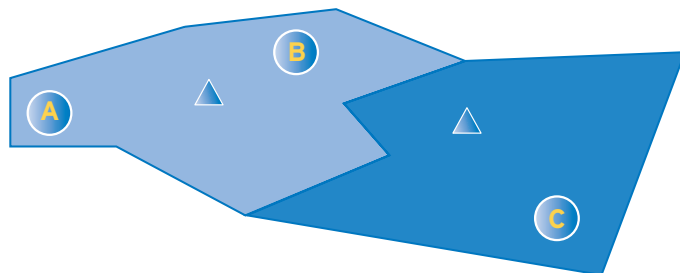


Figure 12: Points within postal-code areas are treated as if located at the area's centroid

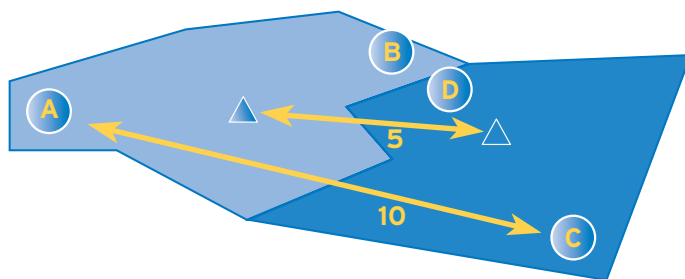


Figure 13: Calculating travel time using postal-code areas

All points within the same postal-code area are considered to be quite close to each other, so that travel between them is assumed to require no driving at all, or a constant small number (e.g., half a mile or less).

Travel from any point in postal-code area 1 to any point in postal-code area 2 is considered to have the same distance, no matter where these points are in within their respective areas. This distance is taken to be the distance from the centroid of area 1 to the centroid of area 2. This is shown in Figure 13, where the distance between the centroids is 5 miles. As a result:

- Distance between A and B will be considered 0 or a constant small number.
- Distance between A and C will be considered 5 miles (even though they are 10 miles apart).
- Distance between B and D will be considered 5 miles (even though they are very close)

<sup>7</sup> This is due to the same reasons discussed here – the postal service prefers to avoid sending its employees to routes that span both river banks.

Travel time is calculated by dividing the distance by a single speed parameter (mph), since this method does not utilize any knowledge of the speed of any actual roads linking the areas. At best, this method might include both travel time and travel distance for each pair of areas but, as seen in Figure 13, this is still not a good match to real drive times.

Figure 14 shows another example of how the simplifying assumptions described here can lead to highly inefficient scheduling. The figure shows four points, all within one postal area. Using postal code resolution, all routes going through points A, B, C and D seem equally good. For example, A → C → B → D seems as good as A → B → C → D. However, if we examine the actual locations of these points, it is obvious that A → B → C → D can save approximately 10 miles and 30 minutes over A → C → B → D.

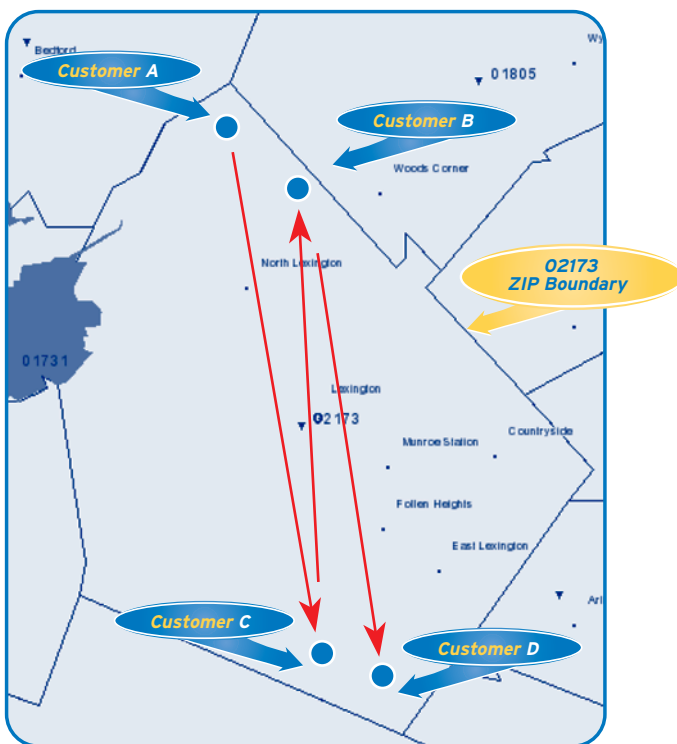


Figure 14: Using postal-code-based travel distances leads to inefficient routes, since distances within the same area are assumed to be constant.

Sizes of postal-code areas vary by large factors between different geographical areas. In some dense urban areas, a postal code – such as the extended “ZIP + 4” codes used in the USA – might extend for only a few city blocks. While this might increase the accuracy, it can do so only if a very high price is paid: precalculating and storing an extremely large number of area-to-area travel information, and keeping it current as roads are added or changed. Furthermore, travel within such dense urban areas is notoriously slow and difficult, so the situation seen in Figure 13 and Figure 14 is repeated in miniature.

### 4.3 Optimize one route at a time

Consider the following two-step approach:

- 1 Use straight-line distance estimates to assign tasks for each mobile worker.
- 2 After these decisions are made, review each worker in turn, and use SLR to determine more realistic drive times between the tasks assigned to that worker. In the process, the order of tasks might be reshuffled to avoid obstacles and take advantage of proximity of tasks to routes between other tasks (as in example 3) or of faster roads (as in example 4).

The rationale for this approach is that compared to optimization across the workforce, far fewer SLR calculations are required. Recall the example of the work group having 120 mobile workers who need to handle about 800 tasks per day, so that each worker has six or seven tasks per day. Since we’d now be calculating distances only within each worker’s assigned tasks, there might be 40 computations per each worker or 4800 (120\*40) computations overall. Recall that optimization across all the tasks and the whole workforce requires over 600,000 SLR calculations.

This way, the problems of “bumping the schedule” due to misestimated travel times are no longer a threat, since the final schedule does use SLR-based estimates within each route. This means that the schedule is realistic. However, the schedule generated by such approaches is incomplete and inefficient.

- **Incomplete:** If the real travel times are significantly larger than the straight-line-based time estimates used in the first step, the workday will not be long enough to handle all the tasks. The tasks that now “fall off the Gantt chart” require either manual fixing (hard to do, since all the workers’ time is fully booked), payment for overtime or postponing some tasks to tomorrow.

## ■ Inefficient:

- In example 1 (having to cross the bridge), SLR will only make sure to allocate enough time for the travel across the bridge. This is more realistic than the original assumption that the proximity of the two points makes it easy to get from one to the other. However, the schedule is still quite inefficient. If the technician had stayed on one side of the river, leaving the other side to be handled by other technicians, productivity could have been much higher. Example 2 shows a similar situation.
- As examples 3 and 4 show, if we utilize real travel times when we make the decision of which tasks should be assigned to which mobile workers, we can squeeze more tasks into each technician's day. If real travel times are only calculated after the decision has been made, we lose these productivity-enhancing opportunities. As Figure 8 vividly shows, once the assignment is cast in stone the best improvement achievable by SLR working separately on each route is just not good enough.

Note that such schedules also reduce the organization's confidence in the scheduling software's decisions, and thus reduce its chances of successful deployment.

## 4.4 Precalculated travel between all known service locations

This approach is not an alternative to true SLR: It is a shortcut towards achieving true SLR.

What if we could precalculate the drive distances and drive times between all locations where we might need to provide service, using SLR to maximize the accuracy? If we do so, the large and time-consuming computation becomes a one-time effort, and every time we need to optimize the schedule we can refer to the stored results.

This idea is well within the definition of "True SLR," **provided that the following conditions hold:**

- Service locations are indeed known in advance.
- The road map does not undergo frequent changes.
- The number of service locations is not excessive.

The last point requires some numerical examples. If a company services capital equipment that is only installed at 1,000 facilities, then it may be worthwhile to precalculate 1,000,000 (one million)

different task-to-task distances. However, if a similarly-sized company services consumer goods that may be owned by any of the 100,000 households in some urban area, the number of required calculations reaches 10,000,000,000: If we have 100 computers, each calculating 20 distances per second, we'd need about half a year to crunch through this – and then repeat for all other urban areas where the company operates. Now consider that during these six months, the road map might be updated – e.g., to fix an error or reflect the opening of a new overpass – thus rendering a large part of the stored calculations irrelevant.

For those companies that meet the above conditions, precalculation is indeed a good idea, and it is fully supported by ClickSoftware's offering.

## 5 Yes, but... Some questions and comments

### 5.1 ...but doesn't it all average out?

One might think that the overestimated travel times in one part of the route compensate for the underestimated travel times in other parts of the same route, so that, on the average, it all works out.

This conception turns out to be false. This is partly because the size of the estimation error tends to be similar for points in the same area type – e.g., mountainous areas might have many large underestimates and only a few overestimates; urban suburbs might have relatively smaller underestimates; and urban centers could conceivably have large underestimates.<sup>8</sup> Since tasks assigned to one mobile worker in one day tend to cluster into similar types of areas, most of the travel segments within each specific route would be either over- or underestimated. Therefore, the total length of these segments would not show any compensation or averaging-out. Instead, the total error would be larger than the error for each individual segment.

It may well be that there would be averaging-out across the whole workforce. For each mobile worker whose total travel time was underestimated by two hours, there might be another worker whose total travel time was overestimated by two hours. This does not solve anything, however: The first worker would have to work two hours of overtime, while the second worker would have two hours with nothing to do (but he would still be paid for these hours).

<sup>8</sup> This could be the case if we take average travel speed within urban centers to be low, but the streets within that center include some that have faster traffic flows and may thus be utilized for faster travel.

Another problem with the “averaging out” argument is that it is enough to have just one significantly wrong estimate in each route to bring about missed SLAs, “fire-fighting” to fix the schedule, etc., as described under the section “Things that go bump in the Gantt: the impact of wrong travel-time estimates.”

## 5.2 ...but isn't SLR also an estimate?

Clearly, street-level routing does not give guaranteed travel times that are accurate to the nearest minute. The software itself might be using average travel speeds appropriate for the types of each road segment in the route; a van stopped in the middle of the road (“just five minutes...”) might delay the unlucky mobile worker, and so on.

If SLR is also “just an estimate,” exactly like straight-line-based estimates, why should we trust SLR to give more trustworthy, more efficient schedules?

The answer, of course, is that accuracy isn't an all-or-nothing affair. In many cases, going for higher-accuracy estimates makes a huge difference in the outcome, even when considering that these estimates are still not 100 percent guaranteed. In running our family errands we think ahead about the roads we need to travel, and we would find it hard to accept the advice that we shouldn't bother since we can't really predict what the minute-to-minute road situation will be. Why should we accept this advice in running a mobile service operation?

## 6 Conclusion

In considering the right way to calculate travel time, we are leaving out many other factors that play a part in creating good schedules. These include a proper fit of the scheduling solution to the organizational processes, business rules and objectives; the capability to handle complex tasks extending over several stages, performed by several crews of people and lasting several days; forecasting and planning across multiple time horizons – from preparing the workforce for anticipated growth and changes months in advance to effective appointment booking for today and tomorrow.

When considering all these factors together, SLR becomes even more important, since the effects of travel-time-estimation errors of just one leg of one mobile worker's route tend to multiply when taking the wider view, as may be seen in the following examples:

- From the strategic planner's point of view, knowing that SLR can achieve 5 percent additional efficiency might be the deciding factor in validating a growth plan.
- From the service manager's point of view, a whole crew might be kept idle when one crew member is late for a task, inflating the impact of wrongly estimated travel times.

Having gone through several noteworthy way stations in a hopefully efficient and useful route, we return to the central theme of this white paper: Some things that seem obvious really are obvious. As long as we don't have flying cars or Buck-Rogers-style jet packs, understanding the real route between two points is essential to running a smooth, efficient service operation – just like it's essential to running the family errands.

## About ClickSoftware

ClickSoftware (NasdaqSC:CKSW) is the leading provider of workforce and service optimisation solutions that maximise workforce productivity and customer satisfaction while minimising operation costs. ClickSoftware's ServiceOptimization Suite provides an integrated, intelligent solution for automatic, efficient, and effective decisions over the entire service decision making chain. The suite includes reliable customer demand and workload forecasting; strategic and tactical capacity planning; optimized shift planning; daily service scheduling; troubleshooting and repair support; wireless workforce management; and business analytics that connect all organisational levels and functions.

ClickSoftware customers represent a cross-section of industry leaders in the telecommunications, computer and office equipment, industrial equipment, medical equipment, building automation, utilities, and home service sectors. Close to 100 customers around the world have implemented ClickSoftware service optimisation solutions to achieve an optimal level of field service.

### Contact Information:

For more information about ClickSoftware  
visit [www.clicksoftware.com](http://www.clicksoftware.com)  
or email [sales@clicksoftware.com](mailto:sales@clicksoftware.com)

### Offices

**NORTH AMERICA**  
**ClickSoftware Inc.**  
35 Corporate Drive  
Suite 140  
Burlington, Massachusetts 01803  
**Tel** (888) 438 3308, (781) 272 5903  
**Fax** (781) 272 6409

**EUROPE**  
**ClickSoftware Europe Ltd**  
270 Bath Road  
Slough  
Berkshire SL1 4DX  
UK  
**Tel** + 44 (0)1753 511166  
**Fax** + 44 (0)1753 553127

**ClickSoftware Central Europe GmbH**  
Hanauer Landstr. 293 B  
D-60314 Frankfurt am Main  
Germany  
**Tel** + 49 (0) 69 489813 0  
**Fax** + 49 (0) 69 489813 99

**ASIA PACIFIC**  
**ClickSoftware Technologies Ltd**  
34 Ha'Barzel Street  
Tel-Aviv 69710  
Israel  
**Tel** + 972 3 765 9400  
**Fax** + 972 3 765 9401

**ClickSoftware Australia PTY Ltd**  
Level 23, HWT Tower  
40 City Road  
Southgate  
Victoria 3006  
Australia  
**Tel** +61 (0)3 9674 7295  
**Fax** +61 (0)3 9674 0400