3-1-2016

Metro Meals on Wheels Treasure Valley Employs a Low-Cost Routing Tool to Improve Deliveries

Andrew S. Manikas
University of Louisville

James R. Kroes
Boise State University

Thomas F. Gattiker
Boise State University

Metro Meals on Wheels Treasure Valley Employs a Low-Cost Routing Tool to Improve Deliveries

Andrew S. Manikas
Management Department
College of Business
University of Louisville
Louisville, Kentucky
andrew.manikas@louisville.edu

James R. Kroes,
College of Business and Economics
Boise State University
Boise, Idaho
jimkroes@boisestate.edu

Thomas F. Gattiker
College of Business and Economics
Boise State University
Boise, Idaho
tomgattiker@boisestate.edu

In this paper, we discuss a project in which we developed a spreadsheet-based system that interfaces with a no-fee driving-directions application programming interface to quickly and accurately build a travel time and distance matrix and then rapidly determine near-optimal delivery-route schedules using a modified genetic algorithm. To the best of our knowledge, the method we used to create the travel matrix had not been employed previously in an academic study. The tool was tested and refined in a humanitarian setting—a local branch of the Meals on Wheels Association of America (now Meals on Wheels America), an organization that combats hunger and poverty by providing food to individuals who are in need. The tool, which is currently being utilized by Metro Meals on Wheels Treasure Valley, has substantially reduced the time required to plan deliveries and has also reduced the delivery driving times by approximately 15 percent.

Keywords: humanitarian logistics; vehicle routing; technology; community; planning; genetic algorithm.
History: This paper has been refereed.

The effective planning and execution of pick-up and delivery operations is a challenge that innumerable organizations face. Many organizations address this challenge, to quickly determine near-optimal delivery routes, by applying complex and high-priced technological solutions. Such costly solutions are, however, often not feasible for budget constrained not-for-profit and humanitarian organizations. The goal of this study was to develop a no-fee tool suitable for use by humanitarian organizations. The tool that we describe in this paper uses a MS Excel spreadsheet and an application programming interface (API) that connects to services (e.g., MapQuest, Google Maps) that provide travel times, distances, and driving directions at no cost over an Internet connection. Importantly, we tested, refined, and implemented the tool in a real humanitarian setting, that is, at a local branch of the Meals on Wheels Association of America (now Meals on Wheels America), an organization that combats hunger and poverty by providing food to individuals who are in need.

This local branch, Metro Meals on Wheels Treasure Valley (Metro Meals on Wheels in the remainder of this paper) in Boise, Idaho, is currently using the tool to improve the efficiency of its meal-delivery planning and operations. A key differentiator between commercial and humanitarian organizations is the extreme budgetary limitations that these humanitarian firms often face. For Metro Meals on Wheels, purchasing a commercial solution was not an option because of its budgetary constraints; this limited the organization to consider only solutions that could be provided free of charge. Because many humanitarian organizations face similar budgetary restrictions, the ultimate intention of this study is to make the tool freely available to any organization in need of a low-cost solution for complex routing problems.

In *Relevant Humanitarian Logistics Research*, we review the related literature and motivation of this study. We then examine the background and objectives of the case study in *Case Study Site*, and follow this with a discussion of the technical concerns relevant to the case in *Technical Considerations*. *Route Optimization Tool* includes a detailed description of the software tool, its underlying optimization heuristic, and the testing effort prior to deployment. *Deployment at Metro Meals on Wheels* describes the implementation of the tool at Metro Meals on Wheels. In the final section, *Conclusions and Future Research*, we discuss the generalization of the tool to other humanitarian contexts, contributions, and future research.

Relevant Humanitarian Logistics Research

Operations research and related fields have produced numerous tools to improve humanitarian operations, including a number of systems that utilize routing-optimization techniques to improve logistics. For example, Eveborn et al. (2009) implemented a scheduling and routing system to improve the efficiency of a home healthcare delivery program. Mahadevan et al. (2013) developed a software system to improve the meal-delivery logistics at a foundation that provides school lunches in India. In line with these studies, Carnes et al. (2013) developed a tool to improve an air-ambulance operation in Canada. These papers and numerous similar studies repeatedly demonstrate the valuable role that operations research and management science often plays in improving humanitarian logistics and operations.

Key Characteristics of Humanitarian Logistics

The literature, for example, Apte (2010), Day et al. (2012), and Kovács and Spens (2009), identifies many factors that make humanitarian operations particularly challenging, including infrastructure problems, vast scopes, geo-political complications, media scrutiny, donor reliance, and surges in supply and demand. Metro Meals on Wheels shares many characteristics with other humanitarian operations. These include the last-mile (i.e., the costs associated with moving goods from a regional or local distribution hub to the end user) transportation problem, uncertainty regarding supply and demand, utilization of nonprofessional (often volunteer) labor, and a limited information technology (IT) platform on which to build and run applications.

Transportation is a major aspect of humanitarian operations and humanitarian logistics. It represents the second largest overhead expenditure (after staff) in most global humanitarian organizations (Disparte 2007). Vehicle fleet management is a key aspect of effective transportation management; and vehicle routing is a key aspect of effective fleet management in general (Crainic and Laporte 1997, Ghiani et al. 2003) and in humanitarian operations per se (McCoy and Lee 2014). Last-mile costs can represent as much of one-third of transportation costs.

Humanitarian organizations often rely heavily on volunteers for ground delivery of goods and services (Fulton 2011, Liu and Robinson 2013). Managing such workforces differs significantly from managing most paid workforces. Volunteers have highly heterogeneous abilities and qualifications, which may not be known a priori. They have limited time availabilities, and volunteer retention can be a challenge. In addition, humanitarian organizations have limited time and resources for managing (e.g., allocating and scheduling) volunteer labor (Falasca and Zobel 2012).

Thomas and Kopczak (2005) identify a lack of IT resources as one of five critical barriers to effective humanitarian logistics. The issue is compounded by the high costs of many IT solutions, which preclude their use in many humanitarian operations (Ergun et al. 2014). Additionally, many donors prefer their money to be used for immediate, tangible relief supplies instead of for management infrastructure, such as information systems or logistical equipment (Oloruntoba and Gray 2006). Given these IT cost issues, as well as the need for portability, sophisticated commercial applications are out of reach for many humanitarian organizations.

This is an author-produced, peer-reviewed version of this article. The final, definitive version of this document can be found online at *Interfaces*, published by INFORMS (Institute for Operations Research and Management Sciences. Copyright restrictions may apply. doi: 10.1287/inte.2015.0835

Case Study Site

Our case site is a regional branch of the Meals on Wheels Association of America. The Meals on Wheels Association of America is a not-for-profit entity that operates over 5,000 local nutritional programs in the United States. The organization, which distributes meals directly to the homes of older citizens (typically 65 years or older) facing food insecurity, delivers approximately 1,000,000 meals per day to recipients living in all 50 states (Meals on Wheels 2015).

Metro Meals on Wheels Treasure Valley operates one of the Meals on Wheels Association of America's local nutritional programs. Operating within Ada County, Idaho, Metro Meals on Wheels leverages a team of volunteer drivers, who drive personal vehicles, and a few paid staff members to deliver meals to the homes of more than 800 clients each day. The meals, which originate at one of several institutional kitchens, are delivered along 21 routes spreading throughout Ada County's 2,745 square-kilometer expanse.

The operational and logistical challenges faced by Metro Meals on Wheels are consistent with the vehicle fleet-management problem that many international humanitarian relief organizations face. Specifically, the organization is challenged to minimize the planning and delivery time in an environment with rapidly changing demand, drivers who are volunteers with varying knowledge and experience, and limited financial resources. The financial constraints mean that Metro Meals on Wheels has no budget for technology acquisition; the result is an IT infrastructure that is very basic; it includes older personal computers with an Internet connection and has no in-vehicle technology.

Information Gathering

Interviews with program administrators at Metro Meals on Wheels highlighted that the logistical aspects of their delivery operations present two main issues. First, time efficiency is critical to successfully delivering meals. The commodity (cooked meals) is perishable and temperature sensitive. To address this, each driver carries a hot and cold cooler to store the warm and chilled portions, respectively, of the meals. To ensure that the meals arrive at appropriate temperatures, the organization attempts to deliver all meals within 90 minutes after the drivers have left the kitchen locations. Second, timeliness is critical because the delivery drivers are volunteers from the community and each driver has a finite amount of time. Additionally, the daily activity of one of the local organization's two full-time employees was dedicated almost entirely to the task of scheduling and routing deliveries, thus preventing her from engaging in any other activities.

We also conducted interviews with the Metro Meals on Wheels route coordinator to understand the existing routing process and develop a plan to improve the process. The specific stops for a route were predetermined; that is, they were clustered by neighborhoods and limited in number so the route would not exceed drivers' time constraints. The number of stops in an individual route ranged from a low of 5 to a high of more than 30. Once the stops on a specific route were determined, the coordinator used a map to manually sequence the stops. The route coordinator attempted to minimize the total route driving time, but without using any operations management heuristics or tools. Next, the coordinator prepared a document of the route, which consolidated the addresses and delivery instructions for each stop. Finally, the coordinator manually entered the route sequence into an online mapping tool to prepare turn-by-turn driving instructions for each delivery volunteer. This process was repeated daily for many routes because meal recipients were added and removed frequently and some meal recipients did not receive a meal every day of the week.

Criteria for Improved Planning and Delivery Processes

We worked with the route coordinator to identify the ideal characteristics of an improved routing process. We focused the design process on the local organization's priorities of reducing both the delivery driving times and the route-planning time, and then agreed upon the following criteria for improving the process:

- Aggregate the existing planning activities into a single system.
- Automate as many planning activities as possible.
- Require minimal user training.
- Allow routes to be easily saved and modified as the day-to-day stop requirements vary.

- Facilitate the inclusion of delivery information specific to Metro Meals on Wheels' customers (e.g., leave meal outside, knock loudly because customer is hard of hearing, pick up used meal trays)
- Automatically create routes that minimize the delivery driving time; Metro Meals on Wheels deemed time minimization to be more important than distance minimization.
- Automatically prepare a document that combines the turn-by-turn driving instructions and delivery-stop instructions.

Review of Commercially Available Solutions

We conducted a comprehensive examination of existing routing tools before deciding to develop a new tool for this project. We deemed complex mathematical optimization tools, such as CPLEX, unsuitable because of their high costs and the requirement that each problem needed to be coded by a skilled user. Next, we examined several online tools and found that the free versions were too limited; for example, these versions restricted the number of stops permitted on routes or did not allow routes to be saved and reused. The stop limitations eliminated the free tools from consideration, because many of the Metro Meals on Wheels routes include over 30 locations and none of the no-fee solutions that we identified has the capability to handle more than 25 stops. Additionally, none of the free solutions could be customized to include the specific instructions and information required for the organization's meal-delivery process. Fee-based versions of several of the online tools did allow a suitable number of stops; however, the annual license fees for these versions exceeded the budgetary capability of Metro Meals on Wheels.

We also explored the applicability of a spreadsheet-based tool using MS Excel, for which the organization already owned a license. The spreadsheet interface was easily configurable to meet the specific needs of Metro Meals on Wheels; in addition, MS Excel's Visual Basic (VB) for Applications (VBA) functionality could be used to access an Internet API to build a travel matrix. MS Excel's Solver optimization tool was not, however, well-suited for this application because it would require a user to create customized optimization models for each route. As an alternative, we proposed that a standard MS Excel spreadsheet could be used to enter the information related to a route; the route optimization could then be conducted using a VBA procedure, which the authors would custom code and which would shield the users from the complexities associated with configuring MS Excel Solver models.

Technical Considerations

Upon deciding to create a spreadsheet-based tool for our client, we recognized that our solution would need to incorporate a route-optimization tool. The desire for efficient route-planning systems has been widely examined in academia and industry (Hill 1982). Although the determination of optimal solutions for the vehicle routing problem (VRP) within reasonable times is often an unrealistic prospect because of the problem's complexity, a variety of algorithms have been developed to quickly find near-optimal solutions. Early heuristics, such as the savings heuristic (Clarke and Wright 1964) and the sweep method (Gillett and Miller 1974), provided techniques for manually generating good solutions to the VRP.

Interestingly, subsequent research focused specifically on Metro Meals on Wheels' delivery problem. Bartholdi et al. (1983) addressed the problem in a study in which the authors designed a heuristic that was simple to use and cost less than $50 to implement. Their solution, which they developed at a time before the widespread deployment of low-cost personal computing, used a manual planning process that reduced delivery times by 13 percent.

Improved Optimization Techniques

Over the past three decades, the computing power of desktop machines has increased dramatically, while their costs have plummeted. Today, users with the appropriate skills and software can generate near-optimal solutions to the VRP using advanced techniques, including simulated annealing (Kirkpatrick et al. 1983), tabu searches (Glover 1986), and genetic algorithms (GAs) (Tang et al. 1996, Varelas et al. 2013), using desktop computers (Markoff 1990). Although each of these algorithms has specific advantages and disadvantages, we chose to use a genetic algorithm for this project because of its relative coding simplicity and robustness (Beheshti and Shamsudding 2013, Nair and Sooda 2010).

This is an author-produced, peer-reviewed version of this article. The final, definitive version of this document can be found online at *Interfaces*, published by INFORMS (Institute for Operations Research and Management Sciences. Copyright restrictions may apply. doi: 10.1287/inte.2015.0835

Travel Matrix Requirements

To be applicable to real-world problems, a VRP optimization tool requires an accurate matrix that contains the travel distances or times between each possible location pair in a system. Regardless of the method employed, the matrix must be constructed before route optimization can

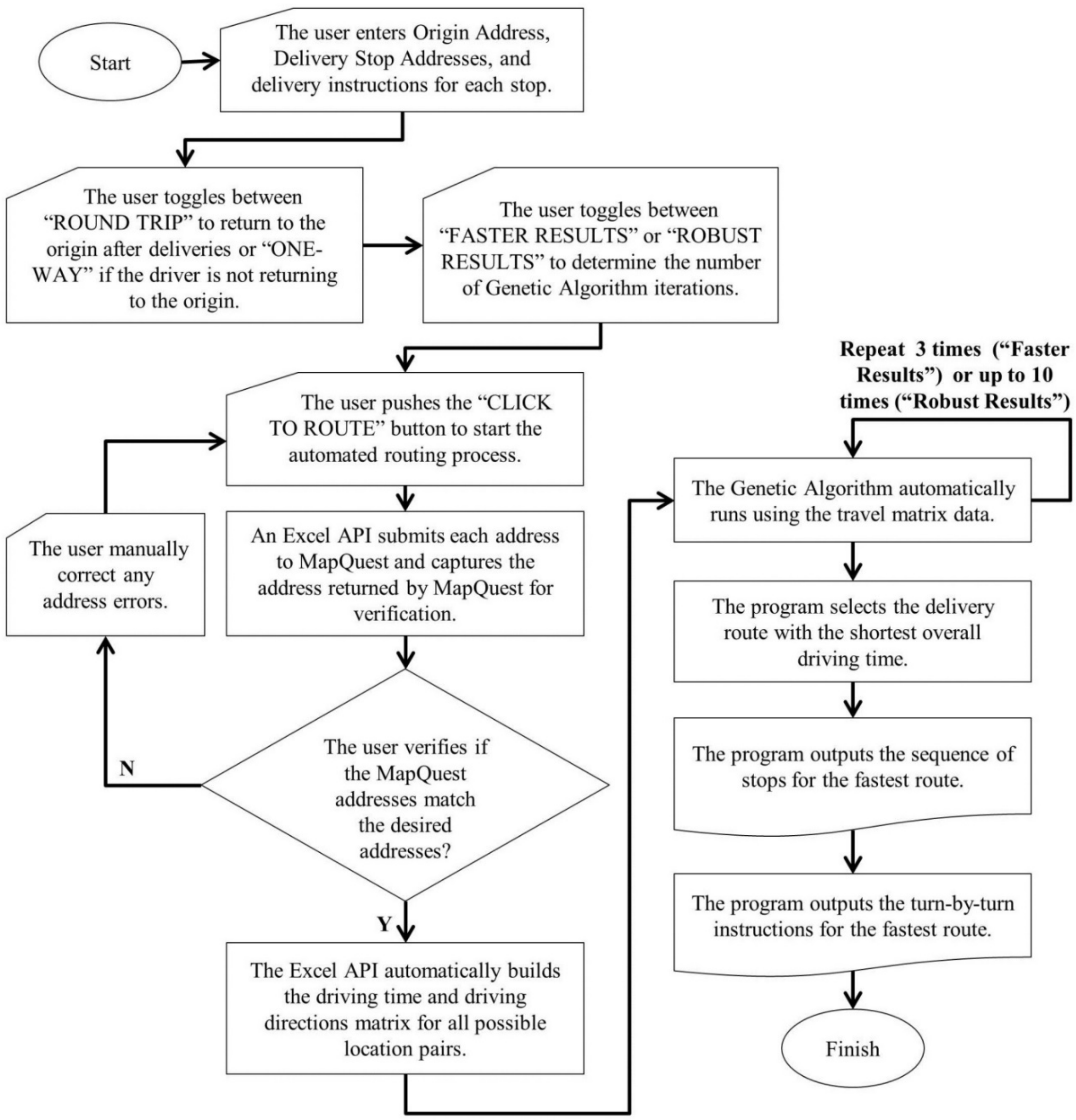

Figure 1: The flowchart describes the operation of the routing tool we developed.

occur. In much of the literature, the travel matrix is often assumed to exist already or it is based on a simple Euclidean coordinate system that assumes linear paths between nodes. In practice, the creation of the travel matrix, which contains the true road distances or driving times between locations, has traditionally required substantial manual effort or the purchasing of data from a commercial provider.

This is an author-produced, peer-reviewed version of this article. The final, definitive version of this document can be found online at *Interfaces*, published by INFORMS (Institute for Operations Research and Management Sciences. Copyright restrictions may apply. doi: 10.1287/inte.2015.0835

Route-Optimization Tool

The tool that we developed has four characteristics; it (1) allows users to input the addresses and delivery instructions for a route, (2) automatically builds a travel-time and driving-directions matrix for the route addresses, (3) utilizes a genetic algorithm to develop an optimal or near-optimal delivery route, and (4) prepares turn-by-turn driving directions and delivery instructions for the route. We developed individual program modules, which we discuss in the succeeding sections, for each of these four functions. Figure 1 presents an overview of the software's operation.

User-Input Interface

For each route, the Metro Meals on Wheels team needed an option to develop both one-way and round-trip solutions, because some drivers must occasionally return to the kitchen after completing their deliveries (e.g., to return equipment, such as coolers and warmers). For each stop on a route, the system needs to track the recipient's name, delivery instructions, meal requirements, and delivery address. These requirements led to the development of the customized user-entry worksheet depicted in Figure 2. We added additional features to allow the user to clear the worksheet and reset the program and to select between "Faster" or "Robust" results, which we explain below. The worksheet was built using standard features available on all versions of MS Excel published since 2010. To operate the program, the user inputs the required data related to a specific route and then simply presses the "CLICK TO ROUTE" button.

CLICK TO ROUTE | **ONE-WAY** | <--Click to toggle between one-way and roundtrip routing
Click to toggle between Faster Results --> or more timely but more Robust Results | **ROBUST RESULTS** | **CLEAR ALL SHEETS**

Assign the Origin location to highlighted row.

Route Name: **Route 1 - Monday**

Stop #	Name	Comments	* No meal left on previous day	# Frozen	Diet	# Hot	# Milk	Address	City	ST	Zip
0	ORIGIN							350 FERST DRIVE NW	ATLANTA	GA	30332
1	John D.	Very Slow to answer door.	*		Gluten Free	1	1	1072 W PEACHTREE ST NW	ATLANTA	GA	30309
2	Mary S.	ph: 555-5555				1	1	1984 HOWELL MILL RD NW	ATLANTA	GA	30327
3	John J.	Please ring doorbell go on in.		1			1	967 BRADY AVE NW	ATLANTA	GA	30318
4	Jane D.	Knock and go on in.				1		570 PIEDMONT AVE NE	ATLANTA	GA	30308
5	James S.	ph: 555-5555	*		Vegan	1	1	190 MARIETTA ST NW STE 280	ATLANTA	GA	30303
6	Dianne J.	ph: 555-5555	*			1	1	1190 N HIGHLAND AVE NE	ATLANTA	GA	30306
7	Jeanette W.	Go to front porch, knock go on in.				1	1	780 MOROSGO DR NE	ATLANTA	GA	30324

Figure 2: The spreadsheet-based user-input interface permits an easy and efficient entry of delivery stops and instructions. (To respect the privacy of meal recipients, we use fictional names and U.S. Postal Service office addresses in this example.)

Constructing the Travel Matrix

A key component of the routing tool is its ability to automatically build a travel matrix for use by the genetic algorithm. Recently, several publically accessible free tools, which allow users to retrieve time and distance information via an Internet API, have become available. For example, Microsoft's Bing Maps API, Google's Maps API, and the MapQuest Developer Network's Map API allow users to access the time and distance information required to build a travel matrix. Of these, we found MapQuest's tool to be the most appropriate for our needs because it allows users to retrieve information for 5,000 location pairs per day free of charge; this is substantially more than Google's limit of 2,500 per day and Bing's limit of 10,000 transactions over a 30-day period.

We construct the travel matrix through the use of a MapQuest API embedded in the MS Excel spreadsheet's VB code. Although we used the MapQuest API for our application, we found Google Map's API to have a more complete location database for many locales outside of the United States. Therefore, we wrote VB coding to support the use of either the MapQuest or Google API; however, we suppressed this functionality for the version of the tool deployed at Metro Meals on Wheels.

When the program begins, the API first queries the entered addresses to allow the user to validate that the actual delivery addresses match the MapQuest addresses. The need for address validation surfaced during early testing when we discovered that the APIs return an incorrect address in some instances in which the address entered does not exactly match the service's version of the address (e.g., entering Main St instead of N Main St may return an incorrect address). Once the addresses are validated, the program then uses the API to retrieve and save the point-to-point driving time, driving distance, and turn-by-turn directions between every possible stop pairing.

We became aware that the driving time between two points (e.g., A to B) is not necessarily the same as the time required to drive between those same two points in the opposite direction (e.g., B to A) because of factors such as one-way streets and traffic configurations. This insight required us to capture information for each possible pair configuration; that is, for a route with n stops, the matrix contains n^2 pairs of information. To decrease the time during future runs, the matrix for a route is not reset each time the program runs; instead, new stop information is appended to the matrix when any new location is added to that route. Additionally, this function allows our tool to be used without any online connectivity once the travel matrix has been constructed.

Route Optimization

Based on Darwin's ideas of evolution, John Holland conceived GAs (Holland 1975). Mitchell (1998) notes that although no single definition of a GA exists, GAs typically consist of at least (1) populations of potential solutions, referred to as chromosomes, (2) selection according to fitness, (3) crossover to produce offspring, and (4) mutation of offspring. The authors programmed the GA they developed for this application as a module embedded in the spreadsheet using MS Excel's VBA language. Although the GA leveraged prior research efforts conducted by the authors, they custom developed the algorithm used in this tool and coded it to meet the specific requirements of this project. The technical aspects of the GA created for this project, which follow Mitchell's recommended format, are discussed in detail in the appendix.

To optimize the stops in a delivery route, the algorithm is first initialized with a population of chromosomes (i.e., various solutions to the problem), each of which consists of a series of stops (analogous to alleles in a chromosome) in the order in which they are to be visited. The total driving time for a specific solution is the sum of the driving time between all stops (alleles) in the route (chromosome). Through an iterative process, incorporating crossover and mutation, the GA evaluates numerous potential driving routes and returns the route with the minimum total driving time. Crossover is accomplished using a 50-50 cross method. Each parent has a 50 percent chance that its next allele (i.e., stop on the route) will be selected. This process goes through all sequences in the chromosome (route), randomly picking the next unique stop from either parent, until a new child chromosome is created with a mixture of stops to form a new route sequence. Mutation is done via pairwise interchange. A chromosome (sequence of stops) is selected, and two stops in that sequence are randomly swapped in their order. If by chance the new sequence violates any precedent relationship constraints, the scoring routine "fitness function," which scores routes based on the total delivery time and feasibility, will eliminate this infeasible sequence. The initial version of the GA followed the typical format of a conventional GA; however, several issues, which we discovered in testing and discuss in the next subsection, required us to modify the GA.

Additional Diversification to Overcome Possible Local Optimal Solutions

During testing, our clients at Metro Meals on Wheels noticed that when they evaluated a specific route multiple times, the program would occasionally produce different solutions. We found that even with a randomly generated initial population and a high mutation rate, the algorithm occasionally intensified toward local optimal solutions. We tested two possible solutions to this problem: one solution increased the population size and mutation rate, and the second solution conducted multiple runs of the GA, each using unique random initial populations. After testing, we determined that the second approach was more effective at overcoming local optimal solutions. We modified the program accordingly to run multiple iterations of the GA; however, because the multiple GA runs extended the program execution time, we added an additional feature to allow users to choose between running the GA either 3 times or up to 10 times. We did this by adding a push button to allow a user to choose between "Faster" results and "Robust" results. When a user selects "Faster" results, three separate GA runs (each with unique randomly generated initial populations) are done and the route with the shortest driving time among the runs is chosen as the best route. When a user selects "Robust" results, the algorithm automatically does up to 10 runs using new initial populations for each run. To decrease the algorithm's run time, a feature terminates the 10-run operation early if 5 of the runs return identical results with the same minimum driving time. When the GA process terminates, the best solution of the runs is kept as the final route. When testing the software with the Metro Meals on Wheels team members, they indicated that although the "Robust" results selection may take more than three times longer than the "Faster" results, they were willing to wait for the "Robust" route solution. Therefore, we removed this feature from the user interface in the version of the program adopted at Metro Meals on Wheels; however, we retained the code for potential use in future versions of the program.

Performance Evaluation

To test the effectiveness of the GA, we optimized a series of test routes to determine if the program generated optimal or near-optimal solutions in reasonable running times. One-way and round-trip solutions were generated for sample routes representing typical small number of routes (5 stops), medium number of routes (7 and 10 stops), and maximum number of routes (30 stops). The 5-, 7-, and 10-stop routes were fully enumerated to determine the optimum stop sequencing; however, full enumeration of the 30-stop route is not feasible on a desktop computer.

When possible, we also evaluated the ability of several publicly available no-fee tools, MapQuest Online, FindTheBestRoute.com, and MyRouteOnline, to compare their optimization of the same routes. As Table 1 shows, our GA found the optimal one-way and round-trip

			Excel genetic algorithm		MapQuest Online tool		FindTheBestRoute.com		MyRouteOnline		Full enumeration	
Test route		Number of stops	Route driving time (MM:SS)	Solution time (MM:SS)	Route driving time (MM:SS)	Solution time (MM:SS)	Route driving time (MM:SS)	Solution time (MM:SS)	Route driving time (MM:SS)	Solution time (MM:SS)	Route driving time (MM:SS)	Solution time (MM:SS)
A	One-way	5	13:42[a]	00:09	13:42[a]	00:04	13:42[a]	00:19	19:00	00:06	13:42[a]	00:02
A	Round-trip	5	23:42[a]	00:11	23:42[a]	00:04	23:42[a]	00:21	32:00	00:11	23:42[a]	00:02
B	One-way	7	37:24[a]	00:16	37:24[a], 41:00[b]	00:05	44:00	00:24	43:00	00:09	37:24[a]	00:03
B	Round-trip	7	43:30[a]	00:17	43:30[a]	00:05	50:00	00:32	52:00	00:10	43:30[a]	00:03
C	One-way	10	14:42[a]	00:34	14:42[a]	00:08	21:00	00:58	Free version is limited to 10 stops		14:42[a]	147:00
C	Round-trip	10	18:57[a]	00:41	18:57[a]	00:08	26:00	00:57	Free version is limited to 10 stops		18:57[a]	155:00
D	One-way	30[c]	62:36	18:33	Tool is limited to 25 stops		Tool is limited to 23 stops		Free version is limited to 10 stops		Was not solved within time limit	
D	Round-trip	30[c]	71:55	18:50	Tool is limited to 25 stops		Tool is limited to 23 stops		Free version is limited to 10 stops		Was not solved within time limit	

[a] Optimal solution.
[b] MapQuest solutions varied depending on the starting order of locations.
[c] The optimal solution for a 30-stop problem could not be determined within a reasonable time limit.

Table 1: The comparison of our tool with commercially available solutions demonstrates our tool's ability to quickly determine near-optimal delivery routes.

routes for the routes with 5, 7, and 10 stops. Of the three commercial solutions tested, only MapQuest Online matched this result. Although MapQuest Online generated solutions more quickly, we found that in one case the optimal route was not returned consistently when the entry order of the stops was modified. Despite MapQuest Online's encouraging results, the tool's 25-stop limitation and inability to capture stop-specific delivery instructions rendered it unsuitable for implementation at Metro Meals on Wheels.

Driver Route Instructions

A vital program feature requested by Metro Meals on Wheels was the automation of the driver route-instruction creation process. As we mention above, one of the route scheduler's most time-consuming tasks prior to this study was the manual consolidation of the route driving information and the stop-specific delivery instructions. To automatically create the instructions for a route, the program progresses through the stop sequence (determined by the GA) and retrieves the turn-by-turn driving instructions between each stop from the travel matrix. The driving directions between each stop and the specific delivery instructions for each stop are compiled in a worksheet to produce a single delivery sheet for a route. Figure 3 presents a sample excerpt of the driving instructions produced by the program.

Field Testing

Before releasing the tool to Metro Meals on Wheels, we conducted field tests to personally assess its effectiveness and usability. We chose a route with a medium number of stops (i.e., 9 stops) and a larger 21-stop route for these tests. To develop baseline performance measures, we first drove both routes following the route sequence that the route coordinator had generated manually. Next, we optimized both routes using the GA and drove them following the driving instructions generated by the

ROUTE NAME: ROUTE 1 - MONDAY	Total route time of 38.42 minutes.	
	---------- T U R N B Y T U R N D I R E C T I O N S ----------	
Leave origin: 350 FERST DRIVE NW		
	Start out going west on Ferst Dr NW toward Regents Dr NW.	Drive 0.2 miles
	Turn left onto Regents Dr NW.	Drive 248 feet
	Regents Dr NW becomes Means St NW.	Drive 253 feet
	Turn right onto Marietta St NW.	Drive 0.6 miles
	Turn slight left to stay on Marietta St NW.	Drive 0.1 miles
	Turn right onto Brady Ave NW.	Drive 0.1 miles
Arrive at STOP: 967 BRADY AVE NW	Delivery for John J., Frozen: 1, Hot: 0, Milk: 1 INSTRUCTIONS: Please ring doorbell go on in.	
--	--	
	Start out going southeast on Brady Ave NW toward 9th St NW.	Drive 0.1 miles
	Turn left onto Marietta St NW.	Drive 1.9 miles
Arrive at STOP: 190 MARIETTA ST NW STE 280	Delivery for *James S., Special Diet Meal: Vegan, Frozen: 0, Hot: 1, Milk: 1 INSTRUCTIONS: ph: 555-5555	
--	--	

Figure 3: The driver instructions generated by the tool combine information that had been previously presented to the volunteers in separate reports.

tool. We found that the tool generated easy-to-follow written directions and route sequences that substantially reduced the delivery driving time. For the nine-stop route, the tool reduced the driving time from 35 minutes to 25 minutes, while also reducing the driving distance from 11 miles to 8.5 miles. The driving time for the 21-stop route decreased from 66 minutes to 57 minutes and the driving distance declined from 21 miles to 18 miles.

Deployment at Metro Meals on Wheels

The tool was deployed at the main Meals on Wheels Treasure Valley office. For this application, a centralized planning location is required to coordinate the delivery logistics, the kitchen operations, and the recipients' specific dietary and nutritional requirements. We met onsite with the route coordinator to load the software on her computer and provide training. We then created a single-page sheet to instruct the route coordinator on the tool's functionality. After approximately 15 minutes of training, she felt comfortable using the tool without our assistance.

To become familiar with the tool, the route coordinator initially tested it on a single route (Shenandoah). Prior to optimization with the GA, the route required 34.3 minutes of driving time. The GA generated a route sequence requiring 27.8 minutes (a 19 percent reduction). Separately, we fully enumerated the Shenandoah route and found that the GA route sequence's driving time was within 0.2 percent of the optimal driving time. The route driver then used the directions generated by the tool to make several deliveries; he reported that they were simple to follow and that the new stop sequence substantially reduced his driving time.

Ten additional Metro Meals on Wheels routes, ranging from 5 to 29 stops, were re-sequenced using the GA tool. As Table 2 shows, the new route sequences generated by the GA resulted in driving-time reductions ranging from 2 percent to 27 percent for each route. Across the 10 routes, the new sequences reduced the daily driving time by 70.5 minutes (or about 5.8 hours across a five-day week). The seven routes containing 11 or fewer stops were also fully enumerated, which confirmed that the GA found the optimal route in all seven instances.

Metro Meals on Wheels implemented our tool in June 2015, and the scheduler used it to optimize approximately 15 of the 21 daily routes served by the organization (the actual number of routes optimized using the tool varies slightly depending on the day of the week). Each day and for each route, the scheduler reviews which customers located on the route will be receiving a meal, updates the stop-location information, optimizes the route, and prepares and prints the turn-by-turn directions and stop instructions for the route driver. The tool is still in use and plans are underway to share it with other Meals on Wheels programs across the country.

When assessing the benefits of this project, it is important to consider both the direct financial benefits and the less tangible benefits. The improved routings reduce the total time spent annually on deliveries by over 530 hours and the driving distance by approximately 10,000 miles. Based on the typical costs to own and operate a vehicle (i.e., $0.58 per mile per year for a mid-size sedan), these savings would reduce the vehicle ownership and operations costs by approximately $5,800 (American Automobile Association 2015). These vehicle cost savings are expenses incurred out of pocket by the volunteer drivers; however, the organization also benefits through improved volunteer satisfaction and retention. Additionally, many of the benefits of the tool are not direct cost savings; hence, they are more difficult to measure in terms of dollar savings. For example, the tool substantially reduced the time spent by the route coordinator in planning the deliveries. The automatic route sequencing and instruction generation takes about 1 minute for medium-sized (10-stop) routes and up to 16 minutes for the larger (29-stop) routes. This compares favorably to the prior manual process, which often required the route coordinator to spend over an hour

Route name	Number of stops	One-way or round-trip	Original route time (minutes)	Spreadsheet genetic algorithm (GA) route time (minutes)	Optimal route time-full enumeration (minutes)	GA vs. optimal route time (% difference)	GA vs. original route time (minutes)	GA vs. original (% difference)
Feds	10	One-way	42.5	31.1	31.1	0.0	-11.4	-27
Garden City	5	One-way	18.2	17.9	17.9	0.0	-0.3	-2
Gekeler	8	One-way	26.2	24.1	24.1	0.0	-2.1	-8
North End	7	One-way	30.6	29.1	29.1	0.0	-1.5	-5
River	7	One-way	20.0	18.0	18.0	0.0	-2.0	-10
Route 1	19	Round-trip	62.3	54.2	-	-	-8.1	-13
Route 3	28	Round-trip	98.2	86.2	-	-	-12.0	-12
Route 4	29	Round-trip	76.9	62.4	-	-	-14.5	-19
Shenandoah	9	One-way	34.3	27.8	27.7	0.4	-6.5	-19
St. Michaels	8	One-way	27.2	21.3	21.3	0.0	-5.9	-22
Trinity	11	One-way	25.4	19.2	19.2	0.0	-6.3	-25
Totals			461.7	391.1			-70.5	-15

Table 2: The tool effectively reduced the driving time required to deliver meals, reducing the time volunteers spend driving by an average of 15 percent per route.

planning a single route. The time saved planning does not result in reduced labor costs for the organization; however, the reduction in planning time has benefitted the organization because the route coordinator (one of only two full-time employees in the organization) is now available to assist with other activities vital to Meals on Wheels' operations. Another benefit, also difficult to quantify, is that the drivers using the new system expressed their appreciation for the more efficient route sequences and streamlined delivery instructions. Although volunteer satisfaction is difficult to value, it is critical, because it directly affects volunteer retention (Wisner et al. 2005). Additionally, on some routes, the reduced route times have benefitted Metro Meals on Wheels because it allowed the organization to increase the number of meals delivered by some volunteers on their routes; this represents a capacity increase for the organization, without a corresponding increase in the number of volunteers.

Conclusions and Future Research

This paper demonstrates the efficiency improvements that can be achieved when sophisticated operations management tools are applied to humanitarian logistics problems. Specifically, in this study, we developed a new method to automatically build a travel matrix that, to the best of our knowledge, had not been previously employed in an academic study. The travel matrix is then used by a modified genetic algorithm to improve the delivery stop sequences for routes served by the Metro Meals on Wheels organization. The case study, which utilized actual delivery routes, successfully validated the tool's ability to deliver operational improvements (i.e., it reduced the time required to plan and execute meal deliveries) for a vital humanitarian organization.

Although this paper discusses a version of the routing tool that we customized to meet the requirements of Metro Meals on Wheels, the core functionality of the tool can be generalized to numerous relief and development humanitarian operations contexts. The benefits of route optimization translate to both relief efforts, focusing on delivery speed, and development efforts, in which cost and efficiency become more important considerations (Pedraza-Martinez et al. 2011). In relief situations, Ergun et al. (2014) found that desktop and laptop computers are the most common systems (after paper systems) employed by relief organizations in Haiti. We suspect that this state of affairs also holds elsewhere; such relief organization have available at least basic computer technology, making our solution feasible for use in many disaster relief situations. Although our tool requires an Internet connection to initially build a travel matrix, Internet connectivity is not required after the matrix has been built. This capability, plus the tool's ability to run on virtually any personal computer, will permit a response team members to use the tool even in areas without connectivity if they construct a matrix of likely relief sites before they deploy to remote locations. The tool's ability to build travel matrices using both MapQuest and Google Maps allows it to be employed in approximately 160 countries—a number that will continue to increase as worldwide mapping efforts expand. Similarly, for disaster relief scenarios in which roads may be closed or damaged, the tool can easily be reconfigured to create routes based on the current road conditions in many settings because both MapQuest and Google Maps track the real-time road and traffic conditions in a number of countries; Google Maps currently tracks this data in 45 countries. Although this study examines an urban last-mile scenario with relatively dense delivery locations, there is no restriction to applying the tool to operations across a wider geographic area.

We are currently working on several enhancements to the tool. First, we recognized that its deployment on a mobile device will substantially improve a driver's ability to respond to changes occurring during a delivery (e.g., traffic, cancellation of a delivery). Using several donated tablet computers, we are currently testing the mobile deployment of the tool, and are hopeful that an additional hardware donation will permit the widespread deployment directly to the route drivers. Additionally, Metro Meals on Wheels is planning to share the tool with other local Meals on Wheels programs across the United States. We expect that the tool can be deployed without modification to a number of other local Meals on Wheels organizations. Additionally, we are developing a modified version of the tool for a local food bank to facilitate both its pick-up and delivery operations.

This is an author-produced, peer-reviewed version of this article. The final, definitive version of this document can be found online at *Interfaces*, published by INFORMS (Institute for Operations Research and Management Sciences. Copyright restrictions may apply. doi: 10.1287/inte.2015.0835

Appendix

We streamlined the GA for speed by using the four components suggested by Mitchell (1998). Below, we show the model representation in *Route Model Representation and Objective Function*, the population initialization in *Initialization,* and the evolutionary process used to create subsequent generations and introduce mutations into the population in *Evolutionary Process* and *Genetic Algorithm Parameter Values,* respectively.

Route Model Representation and Objective Function

The coding of the chromosome to accurately represent candidate solutions is critical to the operation of the GA. A chromosome consists of all nonorigin locations on a route in any order. The origin is always the first location; therefore, it is not encoded in the chromosome for parsimony. Similarly, for round-trip solutions, the origin is also the last location for every chromosome. Each allele points to a number location (0 is the origin) corresponding to actual stops (delivery addresses for meals). The chromosome is composed of the set of alleles (stops in sequence). The objective function, described below in *Objective Function*, evaluates these for fitness.

Notation

i the index for the allele (location) of the solution (route) chromosome.

N the number of stops on a route (for a one-way trip, n is the number of locations inclusive of the origin; for a round-trip, n is the number of locations for the one-way trip plus one more stop at the origin).

R a vector of locations $\{l_0, l_1, l_2, \ldots, l_n\}$.

l_0 the origin from which the meal delivery route starts (meals are loaded here). For a round-trip solution, l_n is *always* the origin.

T_{jk} travel time driving from location j to location k.

TT total driving time to visit all stops in potential solution (R).

All nonorigin locations are unique (i.e., used once). For a one-way trip, all locations including the origin must be visited, and visited once only. For a round-trip, the origin must be the first and last destination.

Objective Function

The total trip duration is the fitness score for a solution. The lower the score, the better the solution is considered for the Meals on Wheels route. We determined the driving distance and time between location pairs using the MS Excel-MapQuest API. The objective function evaluates the best component order of R that minimizes the total driving time on a delivery route:

Minimize $\mathrm{TT} = \sum_{i=1}^{n} T_{j,k}$,

where the index i traverses the vector R for a candidate solution.

Initialization

The initial population consists of randomly generated solutions. Because the routing problem is very complex, we did not attempt to employ heuristics such as the savings method to produce initial population members. Randomly generated initial populations induce diversity into the population and are the fastest to create.

A.3. Evolutionary Process

The evolutionary process utilizes the two genetic operators as Huang et al. (2005) describe. (1) Crossover involves selecting two chromosomes and then combining them into a new child solution composed of alleles from both parents. The crossover operator, as we discuss in *Crossover Process*, converges the population toward a particular solution. (2) Mutation, as we discuss in *Mutation Process*, is intended to introduce diversity and to prevent the heuristic from settling at a local optimal solution.

Crossover Process

Parents are chosen using tournament selection (Miller and Goldberg 1995). Two chromosomes are randomly chosen from the population. The fitter of these two chromosomes (i.e., the one with the lower route driving time according to the objective function, which we discuss in *Objective Function*) is kept as Parent 1. Next, two additional chromosomes are again selected randomly from the population, and the more fit of these is kept as Parent 2.

Crossover is completed using a 50-50 split as Manikas and Chang (2009) describe. The two parent solutions (chromosomes) that survived the tournaments above then "mate" to produce a single offspring. Each chromosome has an equal chance for its allele (a location) to be selected in each iteration of the crossover. Crossover consists of n rounds, where n is the number of possible stops.

Mutation Process

Further diversity beyond crossover is added by using a mutation routine. With a random probability, a solution has a clone offspring made. This offspring has two alleles randomly interchanged (i.e., two stops are switched in order).

Genetic Algorithm Parameter Values

The authors selected GA parameters based on their prior research on GAs for other NP-hard domains; their goal was the generation of reasonably good solutions, but fast solution time.

Population size:	50
Maximum iterations:	1,000
Elite percent:	5 percent
Mutation percent:	10 percent
Early stop:	100

If 100 iterations (early-stop parameter) have passed with no improvement in the top-scoring solution value, then terminate the GA prior to reaching the maximum iterations. This is an enhancement to more quickly return a solution to the user, assuming that an improved solution cannot be found by continuing to run additional iterations of this run.

References

American Automobile Association (2015) Annual cost to own and operate a vehicle falls to $8,698, finds AAA. Accessed October 5, 2015, http://newsroom.aaa.com/2015/04/annual-cost-operate-vehicle-falls-8698-finds-aaa/.

Apte A (2010) *Humanitarian Logistics: A New Field of Research and Action*, vol. 7 (Now Publishers Inc., NL).

Bartholdi III JJ, Platzman LK, Collins RL, Warden III WH (1983) A minimal technology routing system for meals on wheels. *Interfaces* 13(3):1–8.

Beheshti Z, Shamsudding SMH (2013) A review of population-based meta-heuristic algorithms. *Internat. J. Advances Soft Comput. Appl.* 5(1):1–35.

Carnes TA, Henderson SG, Shmoys DB, Ahghari M, MacDonald RD (2013) Mathematical programming guides air-ambulance routing at Ornge. *Interfaces* 43(3):232–239.

Clarke GU, Wright JW (1964) Scheduling of vehicles from a central depot to a number of delivery points. *Oper. Res.* 12(4):568–581.

Crainic TG, Laporte G (1997) Planning models for freight transportation. *Eur. J. Oper. Res.* 97(3):409–438.

Day JM, Melnyk SA, Larson PD, Davis EW, Whybark DC (2012) Humanitarian and disaster relief supply chains: A matter of life and death. *J. Supply Chain Management* 48(2):21–36.

Disparte D (2007) The postman's parallel. *Car Nation* 2(April):22–27.

Ergun Ö, Gui L, Heier Stamm JL, Keskinocak P, Swann J (2014) Improving humanitarian operations through technology-enabled collaboration. *Production Oper. Management* 23(6):1002–1014.

Eveborn P, Rönnqvist M, Einarsdóttir H, Eklund M, Lidén K, Almroth M (2009) Operations research improves quality and efficiency in home care. *Interfaces* 39(1):18–34.

Falasca M, Zobel C (2012) An optimization model for volunteer assignments in humanitarian organizations. *Socio-Econom. Planning Sci.* 46(4):250–260.

Fulton K (2011) Take part in disaster relief. Accessed December 4, 2014, http://mhlnews.com/news/take-part-disaster-relief-0401.

Ghiani G, Guerriero F, Laporte G, Musmanno R (2003) Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies. *Eur. J. Oper. Res.* 151(1):1–11.

Gillett BE, Miller LR (1974) A heuristic algorithm for the vehicle-dispatch problem. *Oper. Res.* 22(2):340–349.

Glover F (1986) Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.* 13(5):533–549.

Hill AV (1982) An experimental comparison of human schedulers and heuristic algorithms for the traveling salesman problem. *J. Oper. Management* 2(4):215–223.

Holland JH (1975) *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (University of Michigan Press, Ann Arbor, MI).

Huang GQ, Zhang XY, Liang L (2005) Towards integrated optimal configuration of platform products, manufacturing processes, and supply chains. *J. Oper. Management* 23(3):267–290.

Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Sci.* 220(4598):671–680.

Kovács G, Spens K (2009) Identifying challenges in humanitarian logistics. *Internat. J. Physical Distribution Logist. Management* 39(6):506–528.

Liu C, Robinson P (2013) Better organisation of volunteers in disaster settings is needed: Lessons for all from China. *Australian New Zealand J. Public Health* 37(6):595–596.

Markoff J (1990) What's the best answer? It's survival of the fittest. New York Times (August 29).

Manikas A, Chang YL (2009) Multi-criteria sequence-dependent job shop scheduling using genetic algorithms. *Comput. Indust. Engrg.* 56(1):179–185.

Mahadevan B, Sivakumar S, Dinesh Kumar D, Ganeshram K (2013) Redesigning midday meal logistics for the Akshaya Patra Foundation: OR at work in feeding hungry school children. *Interfaces* 43(6):530–546.

McCoy JH, Lee HL (2014) Using fairness models to improve equity in health delivery fleet management. *Production Oper. Management* 23(6):965–977.

Meals on Wheels America (2015) U.S. Fact Sheet. Accessed October 26, 2015, http://www.mealsonwheelsamerica.org/docs/default-source/fact-sheets/senior-fact-sheet-national.pdf?sfvrsn=2.

Miller BL, Goldberg DE (1995) Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems* 9(3):193–212.

Mitchell M (1998) *An Introduction to Genetic Algorithms* (MIT Press, Cambridge, MA).

Nair TRG, Sooda K (2010) Comparison of genetic algorithm and simulated annealing technique for optimal path selection in network routing. *NCVN-09 9th and 10th Conf.* (KCG College of Technology, Karapakkam, IN), 36–41.

Oloruntoba R, Gray R (2006) Humanitarian aid: An agile supply chain? *Supply Chain Management: An Internat. J.* 11(2):115–120.

Pedraza-Martinez AJ, Stapleton O, Van Wassenhove LN (2011) Field vehicle fleet management in humanitarian operations: A case-based approach. *J. Oper. Management* 29(5):404–421.

Tang KS, Man KF, Kwong S, He Q (1996) Genetic algorithms and their applications. *IEEE Signal Processing Magazine* 13(6):22–37.

Thomas AS, Kopczak LR (2005) From logistics to supply chain management: The path forward in the humanitarian sector. Report, Fritz Institute, San Francisco.

Varelas T, Archontaki S, Dimotikalis J, Turan O, Lazakis I, Varelas O (2013) Optimizing ship routing to maximize fleet revenue at Danaos. *Interfaces* 43(1):37–47.

Wisner PS, Stringfellow A, Youngdahl WE, Parker L (2005) The service volunteer–loyalty chain: An exploratory study of charitable not-for-profit service organizations. *J. Oper. Management* 23(2):143–161.