On the Formulation and Solution of an Emergency Routing Problem

Final Report
METRANS Project 06-03

Juan Casse* Darin Goldstein† Hsiu-Chin Lin‡
Tariq Shehab§

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation, University Transportation Centers Program, and California Department of Transportation in the interest of information exchange. The U.S. Government and California Department of Transportation assume no liability for the contents or use thereof. The contents do not necessarily reflect the official views or policies of the State of California or the Department of Transportation. This report does not constitute a standard, specification, or regulation.

Abstract

In this work, we will identify important variables that contribute to vehicular movement in an emergency environment. In particular, we formulate and pose the Convoy Routing Problem. We suggest a method for modeling the problem and formulate a precise problem statement that significantly reduces the number of variables under consideration. The difficulty of the proposed problem is examined, practical parameters are gathered via extensive literature search, and an algorithm using artificial intelligence techniques for its solution is presented and empirically analyzed via software simulation.

*Department of Computer Engineering and Computer Science, California State University, Long Beach, CA 90840

†Department of Computer Engineering and Computer Science, California State University, Long Beach, CA 90840 (*dgoldste@csulb.edu*).

‡Department of Computer Engineering and Computer Science, California State University, Long Beach, CA 90840

§Department of Civil Engineering, California State University, Long Beach, CA 90840 (*shehab@csulb.edu*).

Contents

List of Tables

List of Figures

1 Introduction

The aftermath of hurricanes Katrina and Rita and the poor response of the governmental emergency services highlight the need for a more efficient and capable post-disaster asset-movement system. A timely and seamless movement of goods (especially consumable and perishable material) and people is difficult during normal circumstances in any large urban area with complex gridlines and a large transportation network. The complexity is magnified when a disaster eliminates some of the routes and/or destroys some of the storage locations. It is, however, exactly in such circumstances when a robust and adaptive plan of transportation is vital in support of the general population.

A solution for the movement of goods that relies on a relatively static condition of roads and services cannot simply be adopted during and immediately after a disaster; the chaotic movement of precious commodities such as ice after Katrina, for example, was clearly unacceptable. An excerpt from a 2005 article by Matt Ryan (Ryan, 2005) illustrates the basic point.

> "Federally funded disaster relief is not used efficiently because officials don't face the same discipline that people face in private markets. The journey of an ice truck providing Katrina relief illustrates FEMA's inability to coordinate and its blindness to cost e ectiveness. After leaving Wisconsin and arriving in Louisiana, the truck was sent to Georgia but then re-routed to South Carolina. After ending up in Maryland, the truck's ice sat stationary for days, while costing taxpayers money and leaving those in need of relief with fewer supplies."

Clearly, an adaptive/intelligent agent is necessary to manage and direct the movement of goods as the operating environment evolves. When developing a goods distribution system, as Ryan notes above, it is desirable to design a system that is not only able to obtain reasonable solutions for given set of circumstances, but it is also able to adapt its solutions rapidly to unforeseen roadblocks (and potentially take advantage of fortuitous occurrences as they happen).

Commonly, systems that address the goods distribution problem come in the form of models that, given inputs that might consist of a goods manifest and a database of road maps, produce a schedule for distribution. Such models are allowed to search for hours or even days for the optimal solution. Of course, in the face of an immediate and catastrophic emergency, the search for optimal solutions takes too long to compute and implement; a perfect solution, though desirable during normal circumstances, is not possible to obtain in real-time given the fluid nature of a post-catastrophic environment.

The technologies for collecting data on road conditions and goods movement have improved significantly over the past several years. It is now possible to access the velocity of most sections of the major freeways and highways in southern California via the Internet (Technologies, n.d.). Many trucking companies now carry GPS navigation systems and/or electronic tracking devices that allow their central handling agencies to have almost instantaneous knowledge of the positions of the entire fleet at once. Police helicopters are now available in every major population center. A single human being (or, sometimes, even a highly qualified team, as Katrina showed) is no longer able to process all that information at once. The data is available; what is now needed is a system for collecting, organizing, and processing it in a real-time adaptive way.

Our first goal in Section 1.1 will be to identify the variables that contribute to a realistic model of post-disaster traffic flow via comparison with previous work. In Section 2, we will then use this information to derive a mathematical problem that can be analyzed using the tools of theoretical computer science. We make a few remarks about the problem formulation and then delve into analysis of the limiting behavior of the problem itself in Section 2.2. We will show that the problem is most likely intractable (Section 2.3). In Section 3, we discuss realistic parameters for the problem. Finally, we suggest various methods for finding approximate solutions to the problem in spite of its intractability using certain forms of artificial intelligence in Section 4.

1.1 Previous Work

There is a substantial body of work on traffic models. See (Klar & Wegener, 1997a; Klar & Wegener, 1997b; Klar *et al.* , 1996; Jula *et al.* , 2005; Jula *et al.* , 2006; Jula *et al.* , 2008; Chardaire *et al.* , 2005) for a small sampling. In a recent paper by Chardaire et al. (Chardaire *et al.* , 2005), the authors introduce and analyze a problem very similar to ours. This section will be a short comparison of our work with theirs and a justification of our assumptions: Section 1.1.1 illustrates how we are able to sacrifice a certain amount of accuracy in the model in exchange for a potential speed-up in data collection and processing, and Section 1.1.2 is a short discussion of complexity issues and the interaction e ect between entities in the model.

1.1.1 Model Simplification

One of the primary objectives of this work is to put together a realistic mathematical model of traffic flow in such a way that the parameters of the problem are easily gleaned from the surrounding environment. In many published works (e.g. (Jula *et al.* , 2005; Jula *et al.* , 2006; Jula *et al.* , 2008)), fairly complex probabilistic mathematical frameworks are developed for these models and then analyzed using the tools of stochastic analysis. In no case is a closed-form solution found for the overall flow; rather, an artificial intelligence technique of some kind is brought to bear on the result of the stochastic model. The complexity of the models and the large number of variables used makes for a relatively lengthy computational process to locate a sufficiently optimal result.

In an emergency environment, when it comes to movement of supplies and people, solution speed is far more critical a parameter than finding *the* optimal solution. In most cases, a fair approximation to the optimal solution (even to within relatively large constant factors) quickly is far preferable to a perfect but late solution. Additionally, in an evolving environment, the parameters of the problem may (and most likely will) change at a moment's notice. Thus, in our mathematical model, we sacrifice several variables that contribute to the accuracy of the model so that we can reasonably guarantee that the data relevant to the problem can be collected and processed efficiently. In particular, we do away with the standard notion of vehicles (cars, trucks, etc.) and group these into "convoys." In an emergency situation, it is realistic to assume that unless lives are clearly and immediately threatened, people will organize themselves into groups that they trust to travel together, thereby maximizing everyone's safety. In fact, the military, an outfit that for the most part *always* functions in an evolving emergency environment, regularly uses convoys to move vital assets from place to place. Chardaire et al. (Chardaire *et al.* , 2005) use this same simplification in their problem statement as well. To illustrate the importance of the problem to the military, Chardaire et al. (Chardaire *et al.* , 2005) note that an enormous amount of time and resources was spent planning the movements of convoys during the intervention in Kosovo in the late 1990's (Cummings, 1999; Chardaire *et al.* , 2001).

1.1.2 Interactions and Complexity

In order to compare and contrast our mathematical model of vehicle movement with another well-respected model, we will consider Ioannou and Chassiakos (Ioannou & Chassiakos, 2001). In the second section of this work, the authors essentially use a modification of Dijktra's single-source shortest path algorithm (Dijkstra, 1959) to come up with an adaptive algorithm for vehicular re-routing. In this section, the complex model (including potentially numerically solving a first-order di erential equation) that they come up with in the first section of their paper is used to determine the weights on the edges of a digraph. Dijkstra's algorithm is then basically used for each vehicle on this digraph to determine the shortest distance to the terminal point. If their algorithm is run on each vehicle in the digraph, assuming there are n total edges and m total vehicles, the running time of their algorithm is $O(n^3m)$ in the worst case. A quick overview of their analysis shows that if the edge weights change fairly often, as they would in an evolving environment, this worst case running time is the most likely running time. There are two points to note.

- Generally speaking, a cubic dependency on a relatively large parameter is not sufficient for a real-time analysis.
- Interestingly, though the authors have thoroughly studied the movement of an individual truck on a roadway, interactions between vehicles are entirely ignored.

We deal with the first issue by drastically reducing[1] the size of the parameter m and replacing the edge weights with static values[2]. Though this simplification reduces the quality of our model, we believe that the philosophy of reasonably trading accuracy for swiftness applies in this case. We will see later on, however, that this "simplification" does not make the problem trivial. On the contrary, the problem we set up will prove to be most likely intractable and will have to be dealt with using artificial intelligence approximation techniques. As opposed to the solution in (Ioannou & Chassiakos, 2001), these techniques have the benefit that, once a feasible solution is decided upon, the algorithm can be terminated as soon as a "good enough" solution has been found; the longer the algorithm is run, the better the solution. Chardaire et al. (Chardaire *et al.* , 2005) make assumptions similar to ours, grouping vehicles into convoys; however, their model is substantially di erent, and we wait until the formal introduction of the problem to di erentiate.

The second point is more disturbing. Anyone who has driven a vehicle in Los Angeles knows that the presence of one or more additional drivers on the roadway is an extremely significant variable. Such an interaction is certainly multiplied when it comes to convoys of vehicles traveling together. One should assume that two convoys traveling on the same roadway in the same direction will interfere with one another and slow each other down; convoys that have to cross each other at intersections will also cause serious disruptions. Seemingly significant interactions of this kind are included in our model.

2 Formulating the Problem

In this section, we will formulate the Convoy Routing Problem (CRP) rigorously[3]. First, we define what it means to be a "convoy" mathematically.

Definition 1 *Assume that a weighted digraph*[4] $G = (V, E)$ *is given, along with a straight-line embedding into the 2-dimensional plane* $\mathcal{R}^2$ *in such a way that any two distinct directed line segments that are not of the form* $\{(a, b), (b, a)\}$ *either intersect in one point or not at all*[5]*. (The weights on the edges need not necessarily have anything to do with the position of the vertices in the plane.)*

A convoy *at any given point in time is defined to be an ordered sequence of points within the digraph* $(a, v_1, v_2, \ldots, v_c, b)$*. (Note that we allow* $c = 0$*. We*

[1] Recall from Section 1.1.1 that we organize our vehicles into convoys, each convoy being a single large entity.

[2] Note that in (Jula *et al.* , 2006; Jula *et al.* , 2008), the edge weights are not static either.

[3] The Convoy Movement Problem (CMP) was introduced in (Chardaire *et al.* , 2005). These two problems are significantly di erent as we note below.

[4] In most cases, it will be possible to assume that this digraph is planar. However, to keep the model as general as possible and to account for the possibility of bridges and tunnels, we allow nonplanarity.

[5] There are numerous algorithms for straight-line embeddings of planar graphs. We cite two here for reference purposes: (Harel & Sardas, 1995; Chrobak & Payne, 1995). To embed a general graph with these restrictions is trivial: Simply place all the vertices of the graph equidistant on the diameter of a circle.

think of a as the rear of the convoy and b as the front.) We must have that $v_i \in V$ for every valid i. However, the points a and b are not required to be vertices of the digraph; they are required to lie along directed edges in E. More rigorously, $\exists (v_i, v_j) \in E$ such that $a \in \{(1-t)v_i + tv_j | t \in [0,1]\}$ and similarly for b, where the linear combination of two points refers to the linear combination of their respective embedding coordinates in $\mathcal{R}^2$.

We now define what it means for two convoys to "overlap." There are two possibilities for convoy overlap. Two distinct convoys can either (a) be simultaneously proceeding along the same directed edge in such a way that the directed intervals described by the convoys in Definition 1 have nonempty intersection or (b) be crossing at a given vertex.

Definition 2 *Two convoys $A = (a_1, a_2, \dots, a_n)$ and $B = (b_1, b_2, \dots, b_m)$* cross *at a vertex v if and only if the following hold:*

1. *$\exists i, j$ such that $1 < i < n, 1 < j < m$, and $a_i = b_j = v$.*
2. *The pair of equations $a_{i+1}t + a_{i-1}(1-t) = b_{j+1}s + b_{j-1}(1-s)$ has a solution such that $0 < s, t < 1$. (Note that the a's and b's are points in the two-dimensional plane; therefore, this one equation is actually shorthand for two distinct equations.)*

The first part of the definition ensures that an intersection does in fact take place, and the second guarantees that the paths do in fact cross.

Definition 3 *Let two convoys A and B be given as in Definition 1, and fix a particular point in time t. We define the* vertex overlap *(which depends on the time t) to be the number of times that A and B cross as in Definition 2.*

Note that Definition 3 allows two or more convoys to leave from a single vertex or arrive at a single vertex without any associated cost. Also, and more importantly, if two convoys meet at an intersection but do not actually cross paths, we assume that there is no interaction that need occur between the two. This is a simplifying assumption that takes away from the accuracy of the model but not crucially.

Definition 4 *Let two convoys be given as in Definition 1, and fix a particular point in time t. We define the* edge overlap *(which depends on the time t) between the two convoys to be the sum of the weights of the intersecting edges and edge pieces of the two convoys.*

More specifically, consider the intersection of two convoys within the weighted planar digraph G. Note that by Definition 1, we can separate the intersection into disjoint pieces (x, y) such that $\exists (v_i, v_j) \in E$ such that $x = (1-t_x)v_i + t_x v_j$ and $y = (1-t_y)v_i + t_y v_j$ for some $t_x \leq t_y \in [0,1]$. Let the weight of edge (v_i, v_j) be $w_{(v_i,v_j)}$. Then the contribution of (x, y) to the sum is $(t_y - t_x)w_{(v_i,v_j)}$.

Note that Definition 4 simply codifies that the overlap between two convoys traveling parallel to each other is equal to the sum of the overlap distance where distance is defined not in terms of the distance between the points in the embedding of the digraph G in the plane but rather in terms of the weights on the edges. Note that the t's in the definition correspond to the percentage of the edge in question that the convoys have both traveled over. We then take a weighted sum of these traversals to get the final result.

We are now in a position to define the main problem of the paper.

Definition 5 *The Convoy Routing Problem (CRP). A problem instance is given as follows:*

1. *Let $C_1, C_2 \geq 0$ be known constants.*
2. *Let $n \geq 1$ be the number of convoys to be routed.*
3. *Let $G = (V, E)$ be a weighted planar digraph, and let $(v_1^s, v_2^s, \ldots, v_n^s)$ (the "start" vertices) and $(v_1^f, v_2^f, \ldots, v_n^f)$ (the "finish" vertices) be two sequences of distinguished vertices such that for all valid i, $d(v_i^s, v_i^f) < \infty$.*
4. *Let $(x_1, x_2, x_3, \ldots, x_n)$ be a sequence of positive rational numbers, representing convoy lengths traveling at speed 1.*

A solution to this problem comes as a sequence of "routing instructions" for each $1 \leq i \leq n$. A "convoy" is represented by a directed interval within the digraph as described in Definition 1. More specifically, a solution to the problem consists of, for each valid i,

1. *An increasing sequence of n nonnegative rational numbers $(t_1, t_2, t_2, \ldots, t_n)$, "start times," such that $t_1 = 0$.*
2. *For each $1 \leq i \leq n$, a directed path within G represented as a sequence of vertices*
$$(v_{i,1}, v_{i,2}, v_{i,3}, \ldots, v_{i,m_i})$$
such that $\forall 1 \leq i \leq n, v_{i,1} = v_i^s$ and $v_{i,m_i} = v_i^f$.
3. *For each $1 \leq i \leq n$, a sequence of positive rational numbers*
$$(s_{i,1}, s_{i,2}, s_{i,3}, \ldots, s_{i,m_i - 1})$$
that represent convoy speeds. We must have, for every valid i and j, $s_{i,j} \leq 1$. (In other words, 1 is the universal speed limit that we allow in our digraph.)

For each $1 \leq i \leq n$, at time t_i, convoy i is released from $v_i^s = v_{i,1}$. When the front or rear of convoy i reaches $v_{i,j}$, it changes its speed to $s_{i,j}$. The speed *of the endpoints (front or rear) of the convoy is used as follows: Assume that an endpoint of convoy i is at $(1 - T)v_i + Tv_j$ for some $(v_i, v_j) \in E$ and some $T \in [0, 1]$ with speed s, and let the weight of edge (v_i, v_j) be $w_{(v_i, v_j)}$. At time $T^* > T$, there are two cases to consider:*

1. *If* $(T + \frac{s}{w_{(v_i,v_j)}}(T^* - T)) \leq 1$*, then the endpoint of the convoy will have moved* $s(T^* - T)$ *units towards* v_j*. The new position of the endpoint is*

$$(1 - T + \frac{s}{w_{(v_i,v_j)}}(T - T^*))v_i + (T + \frac{s}{w_{(v_i,v_j)}}(T^* - T))v_j$$

2. *Otherwise, the endpoint has reached vertex* v_j *within the time interval* $[T, T^*]$ *and changes speeds accordingly.*

Note that the number of edges that the convoy occupies may increase or decrease as time progresses.

Consider two distinct convoys A and B. Let $f_{A,B}(t)$ *be the amount of edge overlap in the convoys A and B within the digraph at time t as defined in Definition 4. At time t, let* $g_{A,B}(t)$ *equal the vertex overlap between A and B as defined in Definition 3. Let* $T \geq t_n$ *be the time at which the graph G is first empty of convoys. The goal is minimize the cost quantity*

$$T + \sum_{\textit{all convoy pairs } A \neq B} \int_0^T [C_1 f_{A,B}(t) + C_2 g_{A,B}(t)]dt$$

Note that we have defined the problem as an optimization problem. We can easily reformulate the problem into numerous other domains. For the equivalent decision problem, we can simply introduce a parameter k and ask whether the final cost quantity can be made less than or equal to k. If we wish to make the problem online, we can specify that certain parameters are made known to the user at particular times during the execution. For example, the convoy lengths might be known in advance but the number n might be made available to the user only at time t_n, i.e. convoys may wish to leave with no prior warning. The edge weights of the digraph G may change during the execution of the algorithm, and so on.

Each formulation will serve its own function. The optimization problem is the standard presentation that most people are familiar with. The decision problem will serve to prove rigorously in Section 2.3 that the problem presented is not only challenging, but most likely intractable. The online variation of the problem is the most realistic formulation, where the user is informed of the changing parameters of the problem as time progresses.

2.1 Model Comparison with Chardaire, McKeown, Verity-Harrison, and Richardson

In this section, we will compare our formulation of the CRP with the other main convoy model, the CMP formulated rigorously in Chardaire et al. (Chardaire *et al.* , 2005). (In this paper, Chardaire et al. describe their model as a "slightly modified version of the specifications given in Lee et al. (Lee *et al.* , 1996).") In (Chardaire *et al.* , 2005), the problem is phrased as a constrained optimization problem. We found several essential di erences between our model and theirs;

we list most of these di erences below. Which model best encapsulates the situations encountered in practice/on the battlefield is a question that is best answered by those with practical/battlefield experience. (Chardaire et al. use the notation u to represent a convoy iterator; where possible, we try to mimic their notation as long as it does not conflict with our own notation introduced above.)

- Chardaire et al. introduce the parameters b^u which are meant to represent the earliest time at which convoy u can begin its movement. In our model, $b^u = 0$ for every u.
- Chardaire et al. introduce the parameters f^u which are meant to represent the latest time at which convoy u is allowed to move in G. In our model, $f^u = \infty$ for every u.
- Chardaire et al. introduce the parameters g^u which are meant to represent the "waiting interval" of convoy u. In other words, if convoy u does not begin traveling at time exactly b^u, then it must wait for a constant multiple of g^u time units before it is allowed to begin moving. In our model, $g^u = 0$ for every u.
- Chardaire et al. use integer units for times, time intervals, and cost measurements. This makes for a final model in which the methods of integer programming might be applied. Our model is allowed to use weights with values in $\mathcal{Q}^+$.
- Though Chardaire et al. assume that the time it takes for the *head* of convoy u to pass through an edge of G may vary depending on the cost of the edge, the *tail* of each convoy will always take constant time w^u (independent of the edge cost) in order to reach a vertex after the head does. Our model makes the assumption that the heads and tails of convoys travel at the same rate: *both* head and tail of the convoy have travel time between vertices dependent on both the speed of travel and edge weights. However, convoys may have di erent lengths x_i.
- Chardaire et al. assume that the weights of the edges might be di erent for each convoy. Each convoy u has its own separate cost function for the weights, namely $C^u : E \to \mathcal{Z}^+$. Our edge weights are assumed to be constant for every convoy.
- Chardaire et al. assume that convoys are able to "block" other convoys from entering a vertex. In fact, once the head of convoy u enters a vertex in the graph G, no other convoy may touch the vertex for another w^u time units. (Recall that w^u is the time that it takes for the rear of the convoy to reach the vertex.) Our model assumes that two convoys may utilize the same vertex, but that two convoys may cross paths only at some predetermined cost.

- Chardaire et al. assume that convoys may not pass each other moving in opposite directions along edges. We assume that all edges are directed; therefore convoys do not interact with each other if they move in opposite directions between two vertices because they will not be traveling along the same edge.

There are other di erences between the two models, and this is by no means an exhaustive list. Though our model requires far fewer specifications, it will turn out that the computational complexity of the decision problem derived from two models are, perhaps surprisingly, the same.

Note that it is easily possible to make our problem more complex or simple by adjusting the problem statement. We have chosen a formulation that attempts to balance realism with simplicity. Our model has far fewer parameters than does Chardaire et al., our cost function is much simpler to state and evaluate, and we do not require a constrained optimization; however, it is possible to make the argument that Chardaire et al.'s model is more general than ours in certain ways.

One final note about the problem statement: We have chosen to use a uniform maximum speed limit on the roadways, namely 1. As everyone knows, this is not a realistic assumption, and, in addition, the maximum speed on the various roadways may change due to changing weather conditions or other problems that arise. We claim that this uniform choice can be made without any loss of generality in the model by allowing variations in the roadway weights. Assume that the maximum speed limit on a given stretch of roadway with edge weight w is $M \neq 1$. (Obviously, it is also true that $M \neq 0$ or the edge could not exist at all.) We can simply adjust the weight on the edge of that roadway by a factor of $\frac{1}{M}$ to $\frac{1}{M}w$. Note that the time necessary to traverse the roadway of length w at speed M is $\frac{w}{M}$, and the time necessary to traverse the roadway of length $\frac{1}{M}w$ at speed 1 is $\frac{\frac{1}{M}w}{1} = \frac{w}{M}$.

2.2 Limiting Behavior

In this section, we will examine the behavior of the optimization problem given in Definition 5 as the parameters C_1 and C_2 approach their limits.

First, note the limiting behavior of the problem as $C_1, C_2 \to 0$ and as $C_1, C_2 \to \infty$.

- As $C_1, C_2 \to 0$, the emphasis on the "overlap" approaches 0. Thus, we care less and less about convoy overlap and more on the quickest route from start to finish. In the case where $C_1 = C_2 = 0$, the optimal solution is trivial: Using a shortest path algorithm (Dijkstra's single-source shortest path algorithm (Dijkstra, 1959), for example), find the shortest path from v_i^s to v_i^f for each valid i. Let every convoy move along this path at the maximum speed and this optimally solves the problem. This seems to be the case of "maximum emergency": if people do not leave the scene of the emergency as quickly as possible, lives will be lost simply via the delay (e.g. the Chernobyl disaster, Three Mile Island, etc.)

- As $C_1, C_2 \to \infty$, we care less and less about the time needed to get to the destination and more about potential inter-convoy accidents. In this case, there is no real emergency to leave, but it is still the goal to get everyone away from the area eventually (e.g. voluntary evacuations from heavy storm areas, etc.). In the limit, this reduces to the problem of finding disjoint paths in the directed graph G (and this will figure heavily into our proof in Section 2.3).

- If $C_1 \to \infty$ and C_2 remains finite, then the problem reduces to whether there exists an optimal schedule in which no convoy ever shares roadway with any other convoy. If $C_2 \to \infty$ while C_1 remains finite, then the problems reduces to whether there exists an optimal schedule in which no convoy need ever cross paths with any other. Clearly, in both of these cases, there exists a schedule in which no convoy ever need interact with any other convoy: Simply route one convoy after another to its destination and only start the next convoy once the first convoy has completely reached its destination. (Below, in Section 4.3, we will refer to this as Algorithm Obvious.) However, determining an *optimal* solution that minimizes the time necessary is a highly nontrivial problem. Obviously, it may not be necessary to wait until the first convoy has reached its destination before starting the second; there may be ways of routing convoys so that no interaction is necessary. This however is an extremely difficult problem, and because the only methods known for solving it as of this writing are the same as those for the more general problem, we only focus on the more general problem.

Ordinarily, C_1 and C_2 will be reasonably large positive values, reflecting both the urgency of the evacuation and the necessity of separating the convoys so as to avoid collisions and accidents. For example, during the Katrina disaster, where C_1 and C_2 would seem to take such a reasonable "middle" value, if the government had set up organized convoy rescue teams to get the people out and away from the a ected area, it is possible that the unfortunate incidents that occurred might have been avoided.

2.3 The Complexity of CRP

In this section, we will exhibit strong evidence that there does not exist an efficient solution to the decision problem version of optimization problem CRP presented in Definition 5. This will be equally strong evidence that all other versions of the problem we mentioned are intractable as well. Recall that the associated decision problem is the simple modification of the optimization version that chooses in advance a parameter k in addition to the other parameters specified in Definition 5 and asks the question: "Does there exist a schedule with cost less than or equal to k?"

Theorem 1 *The decision version of CRP (henceforward, DCRP) is NP-complete.*

This set of NP-complete problems is interesting because it has been shown that if a single one of these problems is solvable in polynomial time, then all are; however, if it can be shown that a single one of these problems is intractable, then all others must be intractable as well (Cormen *et al.* , 1990). It is currently thought by most theoretical computer science researchers that NP-complete problems are intractable though, to date, there has been no proof of this fact. In the over forty years that researchers have been searching, not a single polynomial time solution has been found for a single one of these problems nor has a single intractability proof been presented.

In order to prove Theorem 1, we will require two separate lemmas, Lemma 1 and Lemma 2. The result will be immediate from these two via the principle of language reduction (Garey & Johnson, 1979).

Lemma 1 $DCRP \in NP$

Proof: Assume that a value of k, a valid DCRP problem instance, and a schedule are given. The goal is to verify that the given schedule does in fact evaluate to a cost less than or equal to the given value k in time polynomial in the problem size. The nontrivial part of this proof arises when we consider that the cost function involves an integral. Recall that the cost of a schedule is

$$T + \sum_{\text{all convoys } A \neq B} \int_0^T [C_1 f_{A,B}(t) + C_2 g_{A,B}(t)] dt$$

from Definition 5.

The value T can clearly be determined from the schedule in polynomial time. If we can show that for a single pair of distinct convoys A and B, the value of $\int_0^T [C_1 f_{A,B}(t) + C_2 g_{A,B}(t)] dt$ can be computed in polynomial time from the given schedule, then the result will follow.

Consider any two distinct convoys A and B. Note that the functions $f_{A,B}(t)$ and $g_{A,B}(t)$ are piecewise linear, and once an intersection is detected, determining the slope of the linear curve is easy. We therefore only really need to determine, for these two convoys, at what times they intersect and in what way (i.e. vertex or edge intersection). It may not be immediately obvious that this is a polynomial time process because time proceeds in a continuous way. The problem therefore reduces to the problem of determining all intersections between convoys during all time up until T.

Because the graph is planar and there are numerous line segment intersection algorithms from computational geometry that function in polynomial time (e.g. Chazelle and Edelsbrunner's optimal algorithm (Chazelle & Edelsbrunner, 1992)), we can determine exactly where and how the convoys intersect at any specific point in time. The question is to determine which points in time should these intersection algorithms be run.

Happily, we can restrict ourselves to the following *events*: Either the front of one of the convoys reaches a new vertex or the rear of one of the convoys reaches a new vertex. Between any one of these events, intersections may occur.

It is very easy to determine whether the trajectories of two points traveling at constant speed intersect; note that between these events all relevant points, namely the front and rear of both convoys, are traveling at constant speeds. Thus, the times of each intersection can be detected in polynomial time, and we add each of these intersections to the event list.

Once we determine the times of each relevant event, we order the events in order of increasing time. We can then create the functions $f_{A,B}(t)$ and $g_{A,B}(t)$ in polynomial time easily by noting that each is piecewise linear between the given ordered events. □

Definition 6 *The Directed Disjoint Paths (DDP) problem is defined as follows. We are given a directed graph G and k pairs of nodes $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$. The problem is to decide whether there exist vertex disjoint paths $\pi_1, \pi_2, \ldots, \pi_k$ so that π_i goes from s_i to t_i.*

The DDP was shown to be NP-complete in Karp's seminal paper (Karp, 1975). Schrijver notes in (Schrijver, 1998) that the planar version of this problem was also shown to be NP-complete in the 1975 paper by Lynch (Lynch, 1975). We will refer to the planar version of the DDP as PDDP. It may be interesting for the reader to note that if the value of k in the PDDP is considered fixed (i.e. is *not* a parameter of the problem itself), then the problem is theoretically solvable in polynomial time (Robertson & Seymour, 1995). This implies that there is hope for the DCRP if we assume that the number n is not a parameter of the algorithm but is rather specified in advance. For example, if we limit the number of convoys that our algorithm will ever be required to route to 20, then there is hope that a polynomial-time algorithm may exist.

Though Chardaire et al. in (Chardaire *et al.* , 2005) also used the DDP in their paper to prove the NP-completeness of their decision problem, their proof was somewhat simpler due to the restrictions that their values were only allowed to be integral and they were allowed to specify finite values for f^u, the time by which convoy u was *required* to have reached its termination point.

Lemma 2 $PDDP \leq DCRP$

Proof: Assume that one is given an instance of PDDP, a directed planar graph $G = (V, E)$ and k pairs of nodes $(s_1, t_1), (s_2, t_2), \ldots, (s_k, t_k)$. We transform this into a DCRP problem via the following. Note that because all paths are vertex disjoint, we know that there cannot exist $i \neq j$ such that $t_i = t_j$. In other words, all path must have distinct destinations. Otherwise, we can easily eliminate this trivial case in polynomial time.

For each vertex $v \in V$, we transform v into $v_1 \rightarrow v_2$ such that all inputs to v head into v_1 and all outputs from v head outwards from v_2. This will form a new graph $G' = (V', E')$ from G. See Figure 1.

We can create the following instance of the DCRP. Let G' as created above be the digraph under consideration. Let $C_1 = 3|V'| + 1, C_2 = 0$, and $n = k$ (where k represents the parameter in the PDDP). Let the weight of each edge of G' be 1. For each $1 \leq i \leq n$, we let v_i^s be the terminal vertex (i.e. v_2 in Figure

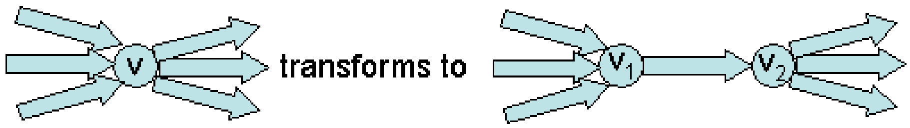

Figure 1: The transformation that occurs for each vertex $v \in V$ to form the new graph G' from G.

1) of the transformed vertex for s_i and we let v_i^f be the initial vertex (i.e. v_1 in Figure 1) of the transformed vertex for t_i. Finally, for each $1 \le i \le n$, let $x_i = 2|V'|$ and let the decision problem ask: "Does there exist a schedule with cost less than or equal to $3|V'|$?" We claim that the answer to this question is YES if and only if the corresponding PDDP question is YES as well. Note that the construction is clearly polynomial-time.

First, assume that the answer to the PDDP question is YES. Then we can route every convoy along its own disjoint path at maximum speed 1. The arrival time of the rear of convoy i is equal to the path length from v_i^s to v_i^f plus the length of convoy i. This expression is uniformly upper bounded by $|V'| + 2|V'| = 3|V'|$. Thus, the schedule performs with cost less than or equal to $3|V'|$ and the answer to the DCRP is YES as well.

Now, assume that the answer to the PDDP question is NO. Let any schedule for the corresponding DCRP problem be given. Let the path taken by the head of convoy i from v_i^s to v_i^f be called π_i. Note that because the answer to the PDDP question is NO, we must have two convoys, say i and j, such that π_i and π_j share at least one vertex $v' \in V'$ in common. Note also that v' will not be the terminal vertex of any path because all convoys have distinct destinations.

Let this schedule run for $|V'|$ time units, and consider the state of the convoys in the network. There are two cases to consider.

1. The vertex v' is not occupied by any convoy at time $t = |V'|$. Note that because each convoy is of length $2|V'|$, each edge is of length 1, and the maximum speed limit is 1, neither convoy i nor j can have already completely passed by the vertex v'. Consider only convoy i: a lower bound for the time that it would take for the rear of convoy i to make it to v_i^f is given by the distance from v' to v_i^f in G' (which is greater than or equal to 1 because $v' \neq v_i^f$) plus the $|V'|$ time units that have already passed by plus an additional $2|V'|$ units to accommodate the entire length of the convoy. This quantity is $\ge 3|V'| + 1$; thus, in this case, the answer to the DCRP must be NO.

2. The vertex v' is occupied by at least one convoy at time $t = |V'|$. Assume that vertex v' is occupied by exactly one convoy at this time. Then by the same reasoning as above, we know that at least one of convoy i or j

must not have reached v'; via the same calculations as above, we know that there exists a convoy that has a lower bound of $3|V'|+1$ units of time before its rear can reach the terminal vertex of its path. Thus, if only one convoy is occupying v', the answer to the DCRP is NO as well.

Finally, consider the possibility that v' is occupied by at least two convoys at time $t = |V'|$. Choose any two of the intersecting convoys, say i and j. At this point, we can note that v' is either the initial or terminal vertex of a transformed vertex from the original graph G.

- Note that at least $|V'| \geq 2$ units of convoy length must remain "behind" each of the intersecting convoys. If v' is the terminal vertex v_2, then we know that at least one unit length of the trailing convoys must be intersecting on the edge $v_1 \rightarrow v_2$ and will continue to intersect completely along this edge for at least one more unit of time after time t. Because $C_2 = 3|V'| + 1$, this will put the cost of the solution over $3|V'| + 1$.
- If v' is the initial vertex v_1, then we know that after one more time unit, the edge $v_1 \rightarrow v_2$ must contain both convoys and will continue to do so for at least one more time unit afterwards because $|V'| \geq 2$. Once again, because $C_2 = 3|V'| + 1$, the cost of the solution must be over $3|V'| + 1$.

In both cases, the answer to the DCRP is NO.

□

3 Realistic Parameters for the Problem

In this section, we perform a literature review on the impact of vehicular speeds given various types of disasters. These values are necessary to determine what the edge weights of the network should be.

3.1 Weather Impact on Speed of Vehicles

Adverse weather conditions have a major impact on the operation of our nation's roads. The impact of weather conditions on highway traffic has been considered an active area of research for many years (Agarwal *et al.* , 2005; Ibrahim & Hall, 1994; Payer & Kuchenho , 2004; Pinelli *et al.* , 2004). Adverse weather conditions such as snow, rain, and wind have been documented to negatively impact traffic characteristics in a number of ways, including vehicular speed. Many researchers have evaluated and quantified this impact and their research findings are presented in the following paragraphs (Chin *et al.* , 2004; FHWA04, 2004; Goodwin, 2004; Kyte *et al.* , 2001).

Goodwin (Goodwin, 2004) performed a study to evaluate the impact of weather on arterial traffic flow. In this study, a number of weather events are

Weather condition	Speed reduction (%)
Dry	0
Rain	10
Wet and snowing	13
Wet and slushy	25
Wheel path slush	30
Snowy and sticking	36

Figure 2: Weather impact (Goodwin (2002))

Weather condition	Speed reduction (%)
Light rain	8
Heavy rain	17
Snow	13–40
High wind	14
Low visibility	15
Combination of snow, low visibility and high wind	30-38
Wet and slush pavement	25
Slushy wheel paths	30

Figure 3: Weather impact (FHWA 2004))

described along with their associated impact on traffic operations. These events are rain, snow, sleet, hail, flooding, high wind, fog, smog, smoke, lighting, and extreme temperature. These events have been documented to influence traffic flow by reducing roadway capacity and speed of vehicles. Speed variability has also been shown to increase as a result of these weather events. Furthermore, the researcher conducted an extensive literature review on the impact of these events on the speed of vehicles. This literature review revealed a speed reduction range of 10% to 36% for various categories of weather events. A detailed description of these weather events and their associated speed reduction values is listed in Table 2.

In a report published by the Federal Highway Administration (FHWA04, 2004), a number of factors were also mentioned to have high impact on the speed of vehicles. These factors are rain, snow, wind, and low visibility. Their impact was found to reduce speed by up to 38%. A detailed description of these impacts and their associated reduction range is given in Table 3.

Chin et.al. (Chin *et al.* , 2004) performed a study to evaluate the temporary losses of highway capacity and quantify their impact on performance. In their study, they categorized highways into four categories: 1) urban freeways; 2) rural freeways; 3) urban arterials and 4) rural arterials. They also identified six weather events of great impact on vehicular speed. These events are light rain, heavy rain, light snow, heavy snow, fog, and ice. The results of this study are

Weather condition	Highway Type			
	Urban freeway	Rural freeway	Urban arterial	Rural arterial
Light rain	10%	10%	10%	10%
Heavy rain	16%	25%	10%	10%
Light snow	15%	15%	13%	13%
Heavy snow	38%	38%	25%	25%
Fog	13%	13%	13%	13%
Ice	38%	38%	25%	25%

Figure 4: Weather impact (Chin et.al. (2004))

Weather condition	Speed reduction (%)
Wet pavement surface	9.5
Snow	16.4
Wind >15 mi/h (24 km/h)	11.7
Visibility < .17 mi (0.28 km)	0.48 mi per .006 mi below 0.17 mi

Figure 5: Weather impact (Kyte et.al. 2001))

shown in Table 4.

As can be noticed from Table 4, the impact of various weather events on urban and rural arterials is exactly the same (i.e. same % of speed reduction). The same observation is also true for urban and rural freeways, except for heavy rain. Heavy rain was found to reduce the speed of vehicles by 16% and 25% on urban and rural highways, respectively.

Kyte et.al. (Kyte *et al.* , 2001) performed a study to evaluate the e ect of weather on free-flow speed of vehicles. In this study a number of weather events were considered: wet pavement surface, snow, wind, and visibility. The impact of these weather events is shown in Table 5.

3.2 Summary of Literature Review

The literature review reveals that there are a number of weather events that negatively impact the speed of vehicles on highways. These events were classified into eleven categories by di erent researchers. These categories are light rain, heavy rain, light snow, heavy snow, wind, low visibility, wet and snowing, wet and slushy, wheel path slush, snowy and sticking and combination of snow, low visibility and high wind. Table 6 lists these events and their associated average impact.

3.3 Road Weather Information Systems

Many states have invested in advanced technologies designed to monitor, report, and forecast road related weather conditions (Boon & Cluett, 2002; FHWA03,

Weather condition	Speed reduction (%)
Light rain	8–10
Heavy rain	10–25
Light snow	13–16.4
Heavy snow	25–40
Ice	25–38
High wind	11.7–14
Low visibility/fog[6]	13–15
Wet and snowing	15
Wet and slushy	25
Wheel path slush	30
Snowy and sticking	36
Snow, low visibility and high wind	30–38
Snow, wet surface, low visibility and wind[7]	Speed = 100.2-16.4 snow-9.5 wet+77.3 vis-11.7 wind

Figure 6: Weather impact (summary)

2003; Decision, n.d.). Collectively, these technologies are referred to as Road Weather Information Systems (RWIS) (Boon & Cluett, 2002). Usually, deployed RWIS components include roadside sensor stations, communication networks, tailored weather forecasting services, advanced weather modeling, pavement temperature modeling and prediction, and an Internet website for decision making and traveler information (Boon & Cluett, 2002). Implementation of these RWIS components serves primarily to enable the use of cost-e ective control practices that improve safety and the level of service provided to users. It should be noted that the road condition and weather information is usually disseminated to the public as a way of helping travelers make informed decisions for safe and efficient travel.

The RWIS usually collect all relevant information using three types of sensors. These are: 1) snow and ice sensors; 2) fog sensors and 3) storm sensors. These di erent types of sensors provide information on water content, density of fog, wind speed and direction, precipitation amount and rate, air temperature, relative humidity and roadway surface conditions (Decision, n.d.). It should be noted that these sensors collect and transmit data to a central processing unit that is usually located in a highway maintenance facility. These central processing units then communicate, collect, archive, and distribute the data.

4 Algorithmic Solutions

By Theorem 1 in Section 2.3, we know that the CRP problem stated in Definition 5 is thought to be computationally intractable by the majority of computer science researchers. In this section, we outline a few alternative possible methods for attacking this problem despite its probable intractability.

Note that there is one relaxation that is immediately obvious. For example, rather than demanding an optimal solution to the problem, one might ask only for a solution that is approximately correct, say to within a certain predetermined factor of optimal. In practical situations, as long as the cost is reasonably bounded, the perfect solution is of secondary importance to its immediate implementation. On this topic, there are several methods utilized in the field of practical artificial intelligence that can most likely be applied. These techniques have implicit bonuses associated with them: As they are run, better and better solutions are located in parameter space; we can stop the calculations at any time with full knowledge of the best solution found thus far. In addition, they are ideally suited for this reason for the online version of the problem. If the parameters are altered in mid-calculation by an adversary (usually, nature), the entire algorithm does not need to be restarted from the beginning.

4.1 Implementation

In order to test whether such an implementation is possible in practice, using the values tabulated in Section 3, we imitated three convoy-routing scenarios with various degrees of disaster intensities on randomly generated maps designed to simulate large sections of a metropolitan area. There are thirty convoys that need to be routed within a region of nine hundred city blocks with thirty randomly generated bridges/tunnels. The epicenter for the disaster is in the center of the city, and the convoys need to be routed around this region. Each section of every street has some probability of being damaged by the disaster.

More specifically, we created 30 convoys; each has a randomly selected length between 2 and 17. Many large cities are designed, for the most part, as grids (e.g. New York City) with various tunnels and bridges. We start our graph with a thirty-by-thirty grid; an example graph is illustrated in Figure 7. The intersections of the grid are vertices in the graph, and each intersection has an edge connected to its neighboring vertices. We add two extra vertices between each adjacent intersection (see Figure 8), thus there are 4380 vertices and 10440 edges (or 5220 two-way streets) in the grid. Every edge is associated with a random weight between 0.7 and 1.3. Recall that the theoretical maximum speed for any convoy is 1 unit. Thus, it takes between 0.7 and 1.3 units of time for the head of a convoy to traverse any given undamaged edge if it travels at maximum speed. The bridges/tunnels are built by randomly adding an edge between two vertices, and the weight of the bridges/tunnels is approximately the Euclidean distance between these two vertices. The starting vertex of each convoy is randomly selected from the lower-left nine-by-nine corner, which accounts for 10 percent of the grid, and the destination vertex is randomly selected from the upper-right nine-by-nine corner (10 percent) of the grid. The epicenter of the disaster occurs at the center of the grid and covers an area of twelve-by-twelve city blocks. A percentage of the streets within the epicenter are damaged, and the rest of the grid sees a damage of one third of that of the epicenter. This percentage is varied between the di erent scenarios. In this experiment, 3 problem instances are generated, and each has 25%, 50%, and 75% of street damage respectively. To add damage to a given stretch of road, we randomly

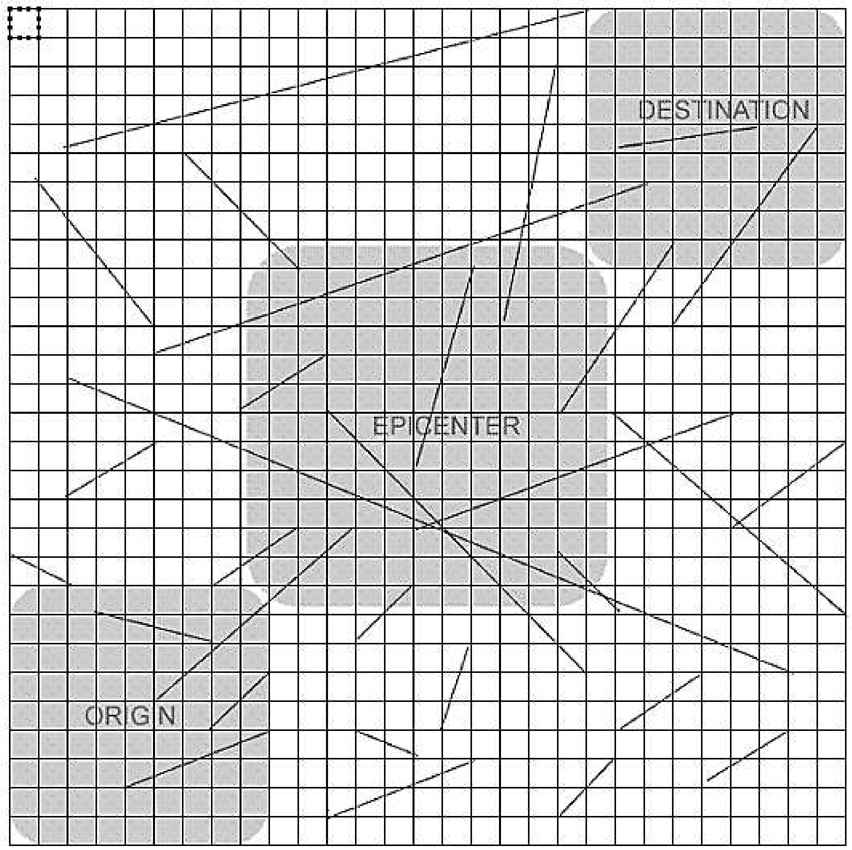

Figure 7: This is a visualization of the imaginary city used as a test case. The bridges/tunnels are randomly generated for each new problem instance. The weight of each edge in the graph is chosen to be 1 plus or minus 30%, and the weight of each tunnel is chosen to be approximately its Euclidean distance. Note the size and location of the convoy origin area, the convoy destination area, and the disaster epicenter.

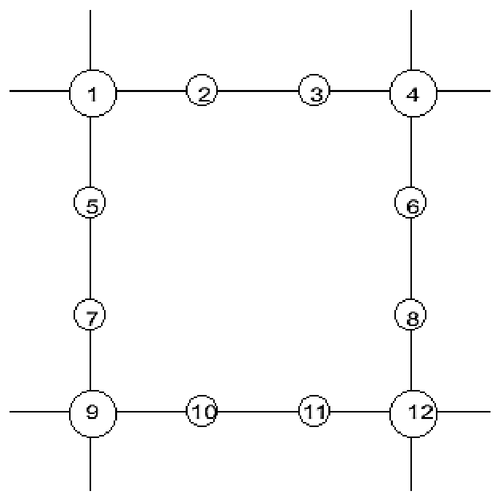

Figure 8: This picture illustrates the additional vertices that we form within each “city block.” This is emphasized in the upper-left corner of Figure 7 as well. Usually a city block will have many more than one “address” per block. In addition to the block intersections, we chose to use the approximation that there are two addresses per city block in order to balance this realistic fact with computational feasibility. (Note that the number of addresses per city block is a multiplicative factor in the number of edges in the network.) In a practical situation, we must assume that it will make no di erence if a convoy that is ten blocks long arrives at an address that is only two doors away from its “real” destination.

select an edge and add a random number between 0 and 13 to its weight.

4.2 The "Original" Genetic Algorithm

Our goal is to find an approximately optimal solution to this problem instance within a time frame that can realistically be called real-time in the context of this problem. The cost of a given solution is computed via discrete-event simulation.

We use a form of genetic algorithm to approximate the optimal solution. To generate an initial solution, we used Dijkstra's single-source shortest path algorithm to find the shortest path for every convoy from their origin to their destination, and each has an initial constant speed of 1 (the theoretical maximum). The initial population consists of 10 copies of the initial solution. For each solution in the population, we randomly select 8 intersection events and avoid them; this creates 8 new solutions.

The algorithm "avoids" intersection events in two di erent ways: (1) slowing down a convoy so as to reach the point where the two convoys intersect just as the other convoy has cleared it and (2) re-routing a convoy. When an intersection occurs, the algorithm chooses one of these methods with probability 50% each.

1. If the algorithm chooses to slow down a convoy, the algorithm will slow down the selected convoy from the vertex prior to the "collision" so that the head of the selected convoy reaches the event vertex just after the tail of the other convoy leaves the collision point.

2. If the algorithm chooses to re-route a convoy, the vertex in the path of the convoy prior to collision is chosen, and the selected convoy is re-routed by finding the shortest path (using Dijkstra's algorithm) from a vertex adjacent to this point to the destination. Note that in the case where the chosen vertex has out-degree two (i.e. the vertex is not at the intersection of the grid), there cannot exist a valid re-routing: any re-route from that vertex must send the back into itself because the only vertex adjacent to the collision is back towards the convoy. In this case, the event is avoided by slowing down the convoy; slowing down a convoy is always possible.

If a derived solution is better than any solution in the current population, the worst solution in the population is replaced by the new solution. Thus, the population always contains the top 10 solutions. The algorithm continues to discover better solutions and eliminate worse solutions until no obvious improvement can be made.

4.3 Evolution of the Genetic Algorithm

Di erent strategies for dealing with event avoidance were attempted. There exists an obvious solution for scheduling the convoys which we refer to as Algorithm Obvious: let the convoys move one at a time; a convoy cannot depart unless the previous convoy arrives at its destination. (Clearly, this is not the optimal situation.) Originally, the genetic algorithm had an equal chance to slow down or re-route a convoy as described above. Our first attempt had final solution cost just slightly less than if we had run Algorithm Obvious. Upon termination, there were still overlapping events, but the algorithm could not derive a better solution by re-routing or slowing down. Our original algorithm performed well on small problems but not on the problem as described in Section 4.1.

Given that the original strategy failed to find a satisfactory solution to our problem, we adjusted the algorithm in the following ways based on our reasoning for why it did not perform well.

1. Instead of re-routing a convoy from the point of collision, we attempted to find a re-route path from the last grid intersection the convoy passed; thus, the algorithm avoids re-routing from an out-degree 2 vertex. Before making this adjustment, it was observed that an unusually large number of convoys were being slowed down rather than rerouted. The reason for this is that a large proportion of the vertices in the graph have out-degree 2. Immediately switching to the slow-down option was not allowing the algorithm to adequately search for potential re-routes. Note that a re-route is now always possible, since a convoy can always be re-routed from its starting point.

2. In the case of an edge intersection, instead of slowing down a convoy on the edge where two convoys start to intersect, we had the algorithm randomly chose a point to slow down from the convoy's path prior to reaching the o ending edge. This way the convoy remains close to its starting point with higher probability keeping clear of other convoys that may be traveling along the same edges, and thus avoiding possible interactions with other convoys.

3. We allowed the algorithm to favor slowing down over re-routing. The original algorithm had equal chance to delay or detour the overlapping convoy, but now, 2/3 of these convoys are slowed, and only 1/3 will be re-routed. Because of the lack of short paths from the origin to the destination in our problem, searching for a quicker re-route that avoids additional intersections will not succeed much of the time. Our algorithm should not have the "patience" to sift through too many failures, but it would be imprudent to ignore the possibility of a successful re-route completely.

After this modification, the second attempt yielded better results. The final cost of the solution dropped orders of magnitude. However, we found that it was possible to increase the performance one more time based on the following observation: If we slow down a convoy in the middle of the grid, that convoy will occupy a certain number of edges while it is "waiting" for the other convoy to pass by the collision point. On the other hand, if the starting time for that same convoy is delayed, as opposed to delaying the convoy while it is blocking up roadways in the network, those otherwise occupied edges would then be available for other convoys to travel or detour. Therefore, delaying the departure time for some convoys is a promising strategy when only a few "short" paths from origin to destination are available as in our case. For the third attempt, the algorithm delays the start time of a convoy during a slow-down if the vertex selected for slow-down happens to be the starting point of that convoy; the start time is changed to a random value not exceeding the maximum amount of time it takes a convoy to reach its destination. (In other words, we at most double the time it takes to reach its destination if it traveled at maximum speed.) It is this final solution for which we present our results below.

The final algorithm then is as follows.

1. Initialization
   (a) Use Dijkstra's algorithm to generate an initial solution
   (b) Build the initial population by making 10 copies of the initial solution
2. Iteration
   (a) For each solution in the population, generate 8 new solutions by selecting 8 random events to avoid. The event avoidance can be performed via two methods:
      i. With probability 1/3, the event avoidance is accomplished via re-routing a convoy, as described above.
      ii. With probability 2/3, the event avoidance is accomplished via slowing down a convoy.
         A. Randomly select a vertex from the path of one of the convoys prior to the intersection point.
         B. If the selected vertex is the origin of the convoy, delay the departure time of the convoy by adding a random number as described above.
         C. If the selected vertex is not the starting point, adjust the speed of the convoy on the path, so that the head of the convoy reaches the intersection immediately after the tail of the other convoy.
      iii. From the 10+8*10=90 solutions in the population, keep the top 10 solutions for the next iteration

Definition 7 *One way to calculate a theoretical lower bound for the cost of* any *solution is to calculate the longest travel time if all convoys take the shortest route at maximum speed from each individual origin to destination* assuming that $c_1 = c_2 = 0$. *We define the* cost ratio *of a given solution S to be the ratio of the cost of solution S to the cost of the lower bound.*

The cost ratio for this experiment up to 30 iterations is shown in Figure 9. For each problem instance we generated, our experiment showed that our final algorithm found the best solution it could find in under 60 iterations,

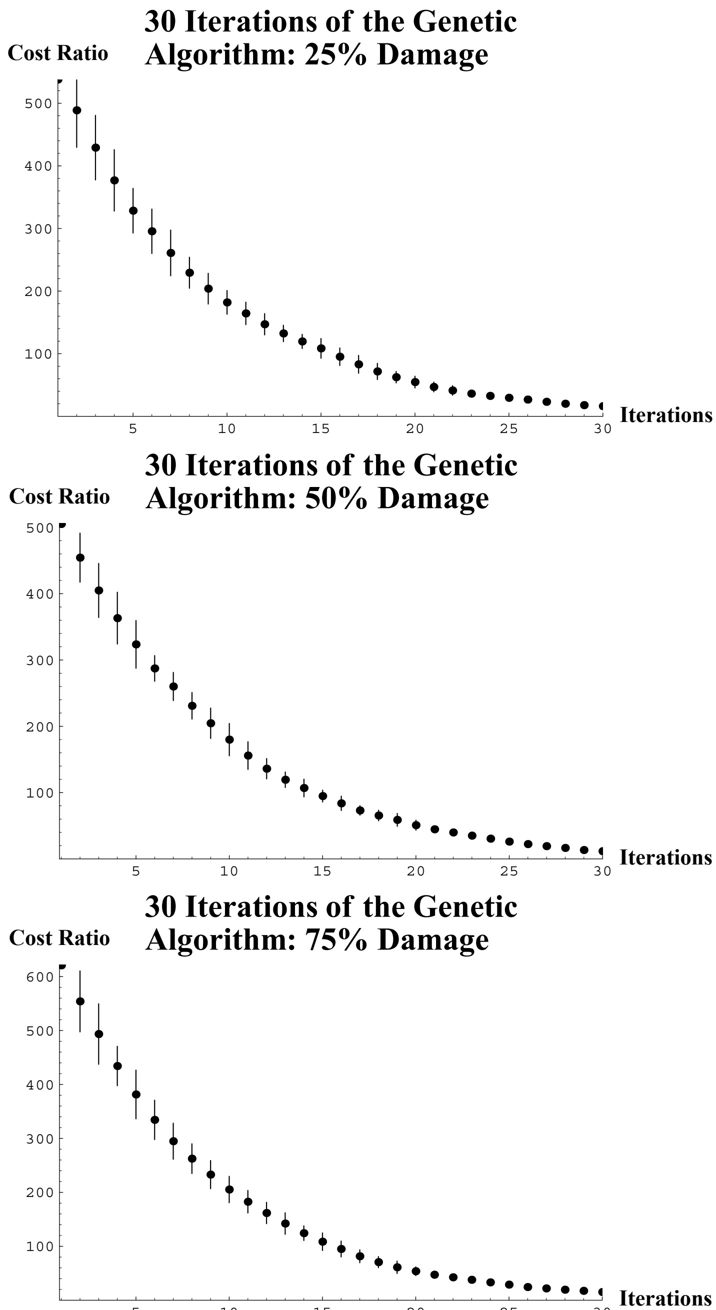

Figure 9: These figures illustrate the results of the genetic algorithm computation. Note that after roughly 30 iterations, in each case the cost ratio is roughly 2, an extremely good result.

and the final cost is *less than twice* the theoretical lower bound on average. Moreover, in some cases, we are able to schedule every convoy from its origin to its destination without any intersection events at all. Each simulation lasted between 5-15 minutes, which, in terms of disaster routing, is close enough to real-time to be acceptable for our purposes.

5 Future Work

Future work includes extending the algorithm to find better solutions more efficiently. Currently, the algorithm terminates when no significant improvement is made after a certain number of iterations or there are no more interactions to avoid. Advancing the convoys' starting time or speeding up convoys may yield better results. Also, instead of randomly picking up an event to avoid, selecting an event that has relatively higher cost might lead to faster convergence.

Other relaxations to the CRP are also possible, perhaps by constraining the C_1, C_2, or M parameters to lie within a certain range depending on the values in G. However, we believe that studying the more general problem is more cost-efficient than pouring resources into researching potentially minor subproblems.

References

Agarwal, M., Maze, T., & Souleyrette, R. 2005. Impacts of weather on urban freeway traffic flow characteristics and facility capacity. *Proceedings of the 2005 Mid-Continent Transportation Research Symposium, Ames, Iowa*.

Boon, C., & Cluett, C. 2002. Road weather information systems: enabling proactive maintenance practices in Washington state. *Washington Research Transportation Center (TRAC), Research Project T1803, Task 39*.

Chardaire, P., McKeown, G.P., Harrison, S.A., & Richardson, S.B. 2001. Convoy planning in a digitized battlespace. *Journal of Defence Science*, **6**(2), 168–175.

Chardaire, P., McKeown, G. P., Verity-Harrison, S. A., & Richardson, S. B. 2005. Solving a Time-Space Network Formulation for the Convoy Movement Problem. *Oper. Res.*, **53**(2), 219–230.

Chazelle, Bernard, & Edelsbrunner, Herbert. 1992. An optimal algorithm for intersecting line segments in the plane. *J. ACM*, **39**(1), 1–54.

Chin, S., Franzese, O., Greene, D., & Hwang, H. 2004. Temporary Losses of Highway Capacity and Impacts on Performance: Phase 2. *Oak Ridge National Laboratory, Report # ORNL/TM-2004/209*.

Chrobak, Marek, & Payne, T. H. 1995. A Linear-Time Algorithm for Drawing a Planar Graph on a Grid. *Information Processing Letters*, **54**(4), 241–246.

Cormen, Thomas T., Leiserson, Charles E., & Rivest, Ronald L. 1990. *Introduction to algorithms*. Cambridge, MA, USA: MIT Press.

Cummings, N. 1999. How OR helped pacify Bosnia. *OR Newsletter (Oper. Res. Soc.)*, **342**, 15–17.

Decision, ITS. Weather detection Systems. *Accessible via http://www.calccit.org/itsdecision/serv_and_tech/Weather_applications/detection.htm*.

Dijkstra, E. W. 1959. A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, **1**, 269–271.

FHWA03. 2003. Best Practice for Road weather Management, final report. *Federal Highway Administration, Report # FHWA-OP-03-081, Washington DC*.

FHWA04. 2004. Identifying and assessing key weather-related parameters and their impacts on traffic operations using simulation, final report. *Federal Highway Administration, Report # FHWA-HRT-04-131, Washington DC*.

Garey, Michael R., & Johnson, David S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: W. H. Freeman & Co.

Goodwin, L. 2004. Weather impacts on arterial traffic flow. *Mitretek Systems, Inc., Falls Church*.

Harel, David, & Sardas, Meir. 1995. *An Algorithm for Straight-Line Drawing of Planar Graphs*. Tech. rept. MCS95-24. Rehovot, Israel.

Ibrahim, & Hall, F. 1994. E ect of adverse weather conditions on speed-flow-occupancy relationships. *Transportation Research Records*, **1457**, 184–191.

Ioannou, P., & Chassiakos, A. 2001. Modeling and Route Guidance of Trucks in Metropolitan Areas. *http://www.metrans.org/research/final/99-07_Final.pdf*.

Jula, H., Dessouky, M., Ioannou, P., & Chassiakos, A. 2005. Container movement by trucks in metropolitan networks: modeling and optimization. *Transportation Research - Part E*, **41**, 235–259.

Jula, H., Dessouky, M., & Ioannou, P. 2006. Truck route planning in non-stationary stochastic networks with Time Windows at Customer Locations. *IEEE Transactions on Intelligent Transportation Systems*, **7**(1), 51–62.

Jula, H., Dessouky, M., , & Ioannou, P. 2008. Real-time arrival time estimation at the nodes of dynamic stochastic networks. *IEEE Transactions on Intelligent Transportation Systems*, **9**(1), 97–110.

Karp, R. M. 1975. On the computational complexity of combinatorial problems. *Networks*, **5**, 45–68.

Klar, A., & Wegener, R. 1997a. *A hierachy of models for multilane vehicular traffic I: Modeling.*

Klar, A., & Wegener, R. 1997b. *A hierachy of models for multilane vehicular traffic II: Numerical and stochastic investigations.*

Klar, A., Kuehne, R., & Wegener, R. 1996. *Mathematical models for vehicular traffic.*

Kyte, M., Khatib, Z., Shannon, P., & Kitchener, F. 2001. E ect of weather on free-flow speed. *Transportation Research Records*, **1776**, 60–68.

Lee, Y.N., McKeown, G.P., & Rayward-Smith, V.J. 1996. The Convoy Movement Problem with Initial Delays. *Pages 213–234 of:* Rayward-Smith, V.J., Oman, I., Reeves, C.R., & Smith, G.D. (eds), *Modern Heurisitc Search Methods*. Wiley.

Lynch, James F. 1975. The equivalence of theorem proving and the interconnection problem. *SIGDA Newsl.*, **5**(3), 31–36.

Payer, T., & Kuchenho , H. 2004. Modelling extreme wind speeds at a German weather station as basic input for a subsequent risk analysis for high-speed trains. *Journal of wind engineering and industrial aerodynamics*, **92**(3), 241–261.

Pinelli, J., Subramanian, C., & Plamondon, M. 2004. Wind e ect on emergency vehicles. *Journal of wind engineering and industrial aerodynamics*, **92**(7), 663–685.

Robertson, N., & Seymour, P.D. 1995. Graph Minors XIII: The Disjoint Paths Problem. *Journal of Combinatorial Theory B*, **63**, 65–110.

Ryan, Matt. 2005. Take the Politics Out of Disaster Relief. *The Independent Institute (Commentary Articles).*

Schrijver. 1998. Routing and Timetabling by Topological Search. *In: ICM: Proceedings of the International Congress of Mathematicians.*

Technologies, Jaytu. *http://www.sigalert.com.*