ABSTRACT

Title of thesis: EVACUATION ROUTE MODELING AND PLANNING WITH GENERAL PURPOSE GPU COMPUTING

David D. Prentiss, Master of Science, 2014

Thesis directed by: Dr. Elise Miller–Hooks
Department of Civil and Environmental Engineering

This work introduces a bilevel, stochastic optimization problem aimed at robust, regional evacuation network design and shelter location under uncertain hazards. A regional planner, acting as a Stackelberg leader, chooses among evacuation–route contraflow operation and shelter location to minimize the expected risk exposure to evacuees. Evacuees then seek an equilibrium with respect to risk exposure in the lower level. An example network is solved exactly with a strategy that takes advantage of a fast, low-memory, equilibrium algorithm and general purpose computing on graphical processing units.

EVACUATION ROUTE MODELING AND PLANNING WITH GENERAL PURPOSE GPU COMPUTING

by

David D. Prentiss

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Master of Science
2014

Advisory Committee:
Dr. Elise Miller–Hooks, Advisor, Chair
Dr. Barton A. Forman
Dr. Steven A. Gabriel

Dedication

To my wife, my mother and especially my father, who always saw an engineer.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

In their review of U.S. national hurricane evacuation practices, Urbina and Wolshon (2003) identified two key issues that regional evacuation planners must address to achieve more effective evacuations: controlling evacuation demand and increasing the capacity of evacuation routes. As the primary means of achieving these goals, they propose sheltering in place to limit the number of evacuees, as well as limited–access, contraflow evacuation routes to exploit existing infrastructure. Numerous works have addressed both of these issues separately. However, no works consider both together as evacuation design options for a single, regional–level planner.

Beginning with Sherali et al. (1991), many works address the shelter–location–allocation problem. Recently, several works including Kongsomsaksakul et al. (2005), Ng et al. (2010), Kulshrestha et al. (2011) and Li et al. (2011, 2012) have posed these design problems within the upper-level decision space for a variety of bilevel problem formulations. Faturechi et al. (2013) and Li et al. (2012) both provide summaries of the related literature. Table 1.1 summarizes the relevant works that adopt a bilevel model to improve evacuation planning.

Contraflow has also been well studied. Urbina and Wolshon (2003) and Wolshon (2001) provide introductions to contraflow during evacuations. Kim et al. (2008) intro-

duce a macroscopic contraflow reconfiguration network problem that proposes an ideal direction for each arc. Chiu et al. (2008) evaluated the effectiveness of contraflow operations for a large–scale evacuation case study.

Recently, Faturechi et al. (2013) considered a building evacuation problem that addressed both the location of structurally hardened locations within a building (i.e., safe rooms), as well as exit and corridor modifications to improve travel times and safety. As such, the context of their work is analogous to a regional evacuation with shelters and contraflow operation. While safe rooms are directly analogous to regional shelters, corridor modifications to a building network are substantially different from the network effects of contraflow operations on highways. Despite these differences, Faturechi et al. (2013) provide an important framework for planning network improvements that reduce risk to evacuees both during sheltering and in transit.

To plan for the optimal use of shelter and contraflow operation options to facilitate evacuations, planners must adopt an optimization objective. Yuan and Han (2010) show, in the context of regional evacuations, that preference among many options is highly dependent on the objective chosen. Most works focus on reducing travel times, while, recently, Yuan and Han (2010), Apivatanagul et al. (2012) and Faturechi et al. (2013) have proposed risk–based performance measures. These objectives are claimed by their proponents to more fully capture the true goals of regional planners.

If some concept of risk exposure is taken as the evacuation design plan's objective, the likelihood and potential severity of one or more hazard scenarios must also be considered. Furthermore, potential hazards may not be uniform with respect to the type and spatial distribution of risk exposure to evacuees. For example, a given shelter may

provide acceptable mitigation of the storm–surge and high–wind risks associated with a hurricane but provide virtually no mitigation of risks associated with a nearby hazardous waste spill. For another, a high–capacity, limited–access, contraflow route may be ideal in the hurricane scenario but would be the source of higher risk exposure if it passed near the site of the hazardous waste spill. Given these uncertainties, a stochastic, multi-hazard problem emerges. Within the context of shelter location and allocation, Li et al. (2011, 2012), and Apivatanagul et al. (2012) consider a stochastic hurricane event, while Faturechi et al. (2013) adopt a true, multihazard approach.

In response to emphasis in the literature and in practice on contraflow operations, sheltering and sheltering in place as the means to improve network performance and safety during regional evacuations, the work presented here develops a bilevel design problem that incorporates those actions as potential service offerings. A novel representation of contraflow design options is introduced that extends current evacuation planning approaches to incorporate potential changes to network topology into the design space. Since the ostensible purpose of regional evacuations is the mitigation of harm to potential evacuees, a risk exposure objective is devised and optimized against an array of heterogeneous hazard scenarios. The resulting design problem is solved with an approach that exploits commonplace, inexpensive computing hardware. This solution strategy is implemented in parallel and takes advantage of relatively recent developments in general–purpose computing on graphics processing units (GPGPU).

Article	Upper–Level Decisions	Upper–Level Objective	Scenario Type	Lower–Level Objective
Apivatanagul et al. (2012)	Evacuee departure time and destination	Risk-based, multiobjective	Stochastic hurricane event	Dynamic user equilibrium
Faturechi et al. (2013)	Building modifications	Min–max risk exposure	Multihazard	User equilibrium and system optimal
Li et al. (2011)	Shelter location–allocation	Minimize cost	Stochastic hurricane event	Stochastic user equilibrium
Li et al. (2012)	Shelter location–allocation	Minimize cost	Stochastic hurricane event	Dynamic user equilibrium
Liu et al. (2006)	Evacuation routing	Maximum throughput	General evacuation	Minimize wait and travel time
Ng et al. (2010)	Shelter allocation	Minimize travel time	General evacuation	User equilibrium
This paper	Shelter location and contraflow routing	Min–max risk exposure	Multi-hazard	User equilibrium

Table 1.1: Summary of relevant works with bilevel evacuation models

Chapter 2: Regional Evacuation Design Problem

Consider a directed graph $G = (N, A)$ with nodes N and links A representing the evacuation zone. The set of nodes represents the physical locations of either link intersections, evacuee origin points, shelters or zone exits. Since the event of an evacuee entering any shelter or exit is equivalent to leaving the network, those nodes are connected to an additional, abstract node that serves as a supersink. The set of links $A = A_1 \cup A_2$ connects the nodes, with A_1 representing physical road segments and A_2 being a set of abstract links representing shelters and connections to various "subgraph reconfigurations" which are described below. Movement of evacuees through the network is represented as flows on the links in A_1, while sheltering and leaving the evacuation zone are represented as flows on the links in A_2.

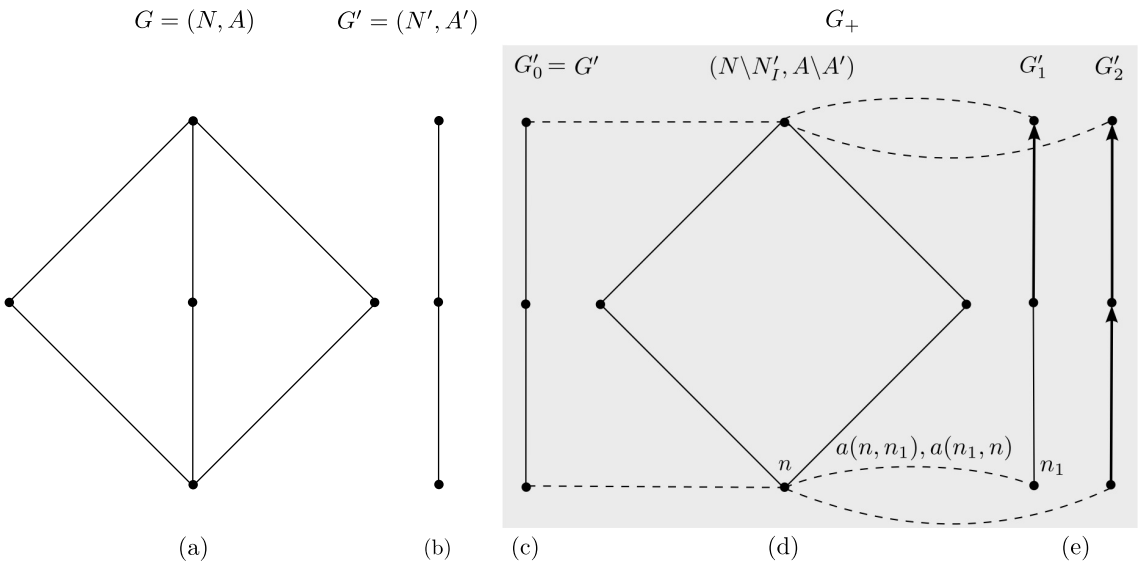

Figure 2.1: The original network (a) and a subgraph (b) are identified for contraflow operation. The original subgraph (b) is removed and reconnected (c) to the surrounding network (d) via abstract links. Two considered subgraph reconfigurations (e) of (b) are also attached via abstract links. The resulting network representation is the union of (c), (d) and (e).

2.1 Contraflow

To model the effects of contraflow operations on network topology, an approach is introduced that results in all design options being represented in a single network. Specifically, a network representation of each considered contraflow option is connected to the network at large via abstract links. The abstract links serve to allow or prohibit flow to their associated network configuration according to the relevant decisions at the upper level.

As an example, consider the process illustrated in Figure 2.1. A segment of highway in the evacuation zone (a) is identified as a candidate for contraflow operation. A subgraph (b) is identified that contains all nodes and links that represent access and direction of traffic on this segment. This subgraph, called here the "original" subgraph, is

copied and abstract links are inserted between the nodes where the original subgraph and the rest of the network intersect. The "interior" nodes of the original subgraph are then deleted along with the adjacent links, leaving its duplicate (c) connected to the surrounding network (d) by the new abstract links. Then, one or more new graphs (e) representing the direction and capacities of the highway under the proposed contraflow options are generated. In the same manner as the original subgraph, the new "subgraph reconfiguration" is connected via additional, abstract links to the intersection nodes. The resulting network representation is the union of the original subgraph (c), the surrounding network (d) and the subgraph reconfigurations (e). This process may be repeated elsewhere in the network until all considered contraflow options are included.

More formally, the network representation of contraflow options is as follows. Consider a connected subgraph of road segments and access intersections $G' = (N', A')$, where $N' \subset N$ and $A' \subset A$.

Definition 1 (subgraph interior node)**.** *An interior node of a subgraph is a node whose neighbors are also in the subgraph. Let* $\Gamma(n)$ *be the set of nodes adjacent to node* n*, the set of interior nodes* N'_I *is*

$$N'_I = \{n \in N' | \forall i \in \Gamma(n), i \in N'\}. \tag{2.1}$$

Definition 2 (subgraph access node)**.** *A subgraph access node is a node that is adjacent to at least one node in the subgraph and at least one other node* not *in the subgraph. The*

set of subgraph access nodes N_T is defined as

$$N_T = \{n \in N | \exists i, j \in \Gamma(n),\ i \in N',\ j \notin N'\} = N' \backslash N'_I. \tag{2.2}$$

Definition 3 (subgraph reconfiguration)**.** *A subgraph reconfiguration G'_r of G' is a connected graph that represents the subgraph G' reconfigured according to some considered contraflow option and possesses nodes N_r and a bijection such that $N_T \rightarrow N_r$. The bijection represents the interface between the subgraph reconfigurations and the rest of the network. By implication, $|N_r| = |N_T|$.*

Any number of subgraph reconfigurations may be generated as part of the formulation, and a copy of G', representing the "do nothing" option, is generated and included among them as subgraph reconfiguration G'_0. For each subgraph G', the resulting network configuration G_+ is

$$G_+ \equiv (N \backslash N'_I, A \backslash A') \cup \bigcup_{r \in R} G'_r \cup \bigcup_{n \in N_T} \bigcup_{r \in R} \big[a(n, n_r) \cup a(n_r, n)\big], \tag{2.3}$$

where R is the set of subgraph reconfiguration indices, n_r is the node in G'_r to which node $n \in N_T$ maps and $a(n, n_r)$ is the abstract link from n to n_r.

To model the case where a contraflow option *is not* chosen (i.e., the decision to do nothing for the subgraph in question), the capacities of the abstract links connecting the contraflow subgraphs ($r \neq 0$) are set to zero, while those of the original configuration ($r = 0$) are set to infinity. As a result, the network in this area is equivalent to the original network. Alternatively, if a contraflow option *is* chosen (e.g., $r = 1$) the capacities of

the abstract links connecting it to the network at large are set to infinity, while those of the remaining abstract links ($r \neq 1$) are set to zero. In this manner, any number of possible contraflow operation options may be considered while maintaining a single network representation through the modeling and solution processes.

2.2 Risk Exposure and Hazards

As users move through the network or take shelter, they are exposed to risks associated with a given hazard. Risk exposure accumulates for each user based on the elapsed time traversing various links of the network and is assumed to remain constant once the user has entered a shelter or has left the evacuation zone.

Adopting the approach of Faturechi et al. (2013), risk exposure is taken as disutility to users and is modeled as a function of the paths chosen in seeking shelter, sheltering in place or leaving the network under a given hazard scenario. The rate at which risk exposure is accrued over time for users on a particular link under scenario ξ is represented by the travel–time risk exposure parameter $\alpha(\xi)$. Risk exposure due to sheltering at a particular shelter under that same scenario is represented by the shelter risk exposure parameter $\beta(\xi)$. This parameter represents the risk to a sheltered user due to all effects of a particular scenario on the relevant shelter for the duration of the event. With these, the risk exposure r_a accrued to a user due to using link $a \in A$ may be expressed as

$$r_a\big[x_a(\xi)\big] = \alpha(\xi) \cdot t_a\big[x_a(\xi)\big] + \beta(\xi), \tag{2.4}$$

where $t_a\big[x_a(\xi)\big]$ is the travel time on link a as a function of flow $x_a(\xi)$ on that link under

scenario ξ. As such, the magnitude of risk on a link is also a function of other users' decisions as a result of congestion delays. Link travel time is modeled by the standard Bureau of Public Roads (BPR) volume delay function

$$t_a[x_a(\xi)] = t_0 \left[1 + 0.15 \left(\frac{x_a}{c_a} \right)^4 \right], \tag{2.5}$$

where t_0 and c_a are the free–flow travel time and capacity of link a respectively.

To choose the values of the risk exposure parameters, modelers may employ any method that captures the relative risks of using each link under some given conditions. The method proposed here is space based and presupposes that, at any given time, the risk to a user in the network is a result of the type and location of a hazard, the user's geographical location and her disposition with respect to transit and sheltering. Hazards in the present context are any events, present or imminent, that prompt an evacuation from the area in question. Furthermore, hazards of practical interest are those under which users face varying risk exposure based on their respective evacuation and sheltering choices. That is, hazards prompting no–notice evacuations (less than 72 hours' notice) in which route choice and evacuation times have a substantial impact on risk exposure are of primary concern.

It is assumed that a hazard implies a location in or near the evacuation zone in which it imparts maximum risk. This "risk epicenter" may comprise any geometry that captures the physical extent of the related maximum risk. A risk epicenter for a hurricane in the present sense might be a circle representing the current location of maximum wind speeds. Or, in the imminent sense, it could be the coastline near the hurricane's expected

landfall and represent the location of maximum risk due to likely storm surge. A risk epicenter for a point hazard might be, as the name implies, a single point. This could model an instance of a single radiological hazard. Alternatively, a group of points could represent the modeled, likely locations of such an event given some imminent threat.

It is further assumed, without loss of generality, that risk attenuates continuously with distance from the hazard epicenter. In the above hurricane example, risk could be modeled to attenuate with linear distance from the coastline. Any mapping, however, from geographical location to risk that captures the modeler's assumptions about space–based risk for a given hazard will serve the current purpose. An example of hazard planning data that lends itself to risk modeling as described above is the National Weather Service's SLOSH Model (National Weather Service, 2014). This service comprises several regional planning products that facilitate storm surge estimates for coastal areas from either deterministic, probabilistic or composite hurricane scenarios. Figure 2.2 depicts a Storm Surge Maximum Envelope of Water (MEOW) result of the SLOSH software. This product maps the predicted storm surge elevation in the region for a given scenario. From this map, regional planners may devise a function or process that transforms these storm surge values to space–based risk parameters.

With a map of risk for a given hazard scenario, the risk exposure parameters may be generated. For a link representing a road segment ($a \in A_1$), users on that link are assumed to be in transit. As such, the travel–time risk exposure parameter $\alpha(\xi)$ is commensurate with the value of the risk attenuation map at that location and the fact that users are in transit and further exposed to whatever extent consideration of that disposition shall require. The shelter risk exposure parameter $\beta(\xi)$, in this case, is set to zero. Since the

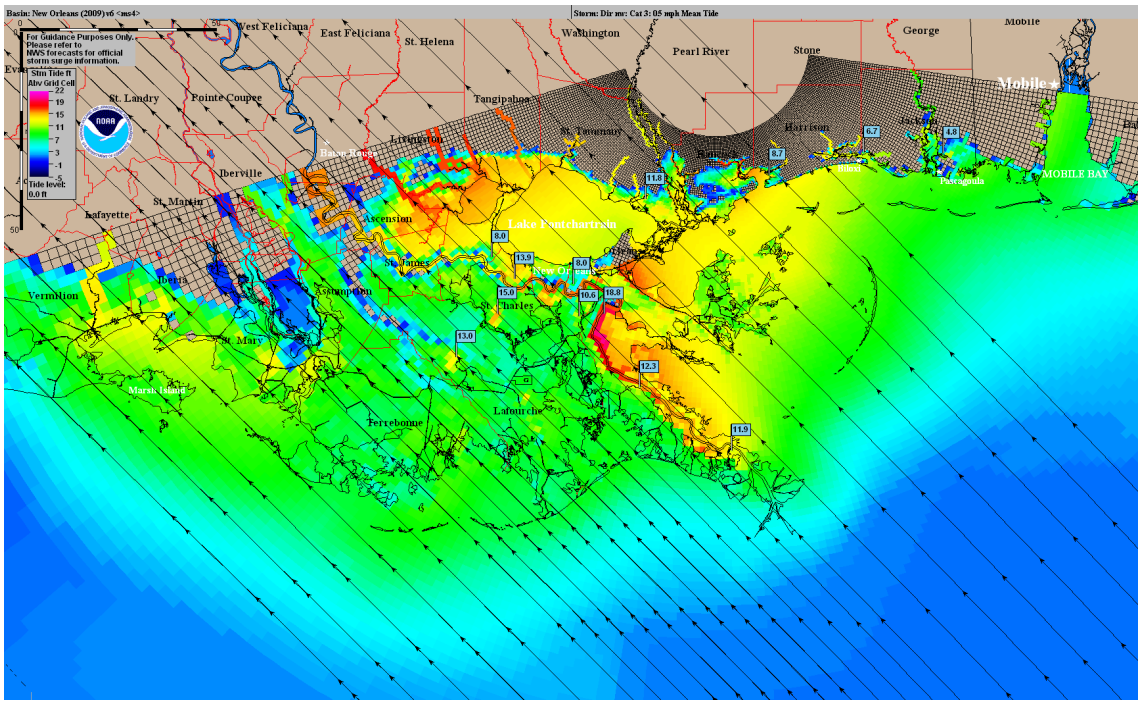

Figure 2.2: National Weather Service (2014). A typical Storm Surge Maximum Envelope of Water (MEOW) map. This MEOW is a typical output of the SLOSH modeling software. It depicts the modeled storm surge for a composite, category 3 hurricane event for the Mississippi Delta region.

represented road segments are extended in space, it must be decided which point or points determine risk for the link. In this work, the relevant point is taken to be the incident node with the greatest risk attenuation value. Alternatively, users on a link that is not a road ($a \in A_2$) are considered to be sheltered or to have left the network.

The value of $\beta(\xi)$ for links representing shelters or sheltering in place is, likewise, commensurate with the value of the risk attenuation map at the location of the tail node of the link and with the level of protection it represents. Since $\beta(\xi)$ is chosen to capture the full, accrued risk exposure due to entering a shelter, $\alpha(\xi)$ is set to zero.

Finally, for exit and connection links, both values are set to zero. Populating (2.4) with these values results in the following link–disutility function:

$$r_a\big[x_a(\xi)\big] = \begin{cases} \alpha(\xi) t_0 \left[1 + 0.15 \left(\frac{x_a}{c_a}\right)^4\right], & \text{if } a \in A_1; \\ \beta(\xi), & \text{if } a \text{ is shelter;} \\ 0, & \text{if } a \text{ is an exit or connection.} \end{cases} \tag{2.6}$$

2.3 Shelters and Sheltering in Place

To model shelters and sheltering in place, each node representing a demand source or potential shelter site is connected to the supersink by an abstract link. Each link has maximum capacity and risk attributes commensurate with user decisions to shelter in place or enter an open shelter. Specifically, the links representing the lower–level decision to shelter in place have capacity attributes equal to demand at that node while shelter

links have capacity attributes equal to the predetermined capacity associated with the shelter–investment decision in the upper level. In both cases, the travel–time risk exposure parameter is zero and shelter risk is predetermined as described above.

Chapter 3: Formulation

As in Faturechi et al. (2013), the Regional Evacuation Design Problem (REDP) is formulated as a bilevel, two–stage, stochastic problem. The upper level represents the decisions and objectives of a regional evacuation planner. These are modeled according to the assumption that the planner, acting as a benevolent Stackelberg leader, will choose among a finite set of predetermined, contraflow operation and shelter location options with the objective of minimizing the maximum expected risk exposure to users under all considered scenarios. The upper–level decisions, taken in the first stage, determine the routes and shelters available to evacuees in the lower level who, in the second stage, choose to shelter in place or choose a route to a shelter or exit. User choices in the lower level are estimated with a user equilibrium of risk–based disutility in which evacuees selfishly minimize their own risk exposure. Furthermore, the lower level takes the form of the capacitated user equilibrium traffic assignment problem (CTAP), introduced by Larsson and Patriksson (1995).

Parameters	
A	set of links, $A = A_1 \cup A_2$
A_1	set of links representing physical road segments
A_2	set of abstract links connecting physical nodes to shelters or contraflow subnetworks
B	total budget for shelter and road segment preparation and management
C	set of contraflow operation options
g_s	cost of investing in shelter option $s \in S$
g_c	cost of investing in contraflow operation option $c \in C$
G	network graph, $G = (N, A)$
K_o	set of simple paths with start node $o \in O$
N	set of nodes
O	set of origin nodes, $O \subset N$
$p_a(y)$	the capacity of link $a \in A_2$ resulting from decision $y \in S$
S	set of shelter options
$\delta^o_{a,k}$	element of the link–path incidence matrix Δ where $\delta^o_{a,k} = 1$ if link $a \in A$ is a link in path $k \in K_o$ origination at node $o \in 0$ and $\delta^o_{a,k} = 0$ otherwise
Δ	link–path incidence matrix
ξ	a scenario, $\xi \in \Xi$
Ξ	set of possible hazard scenarios

First–stage Decision Variables	
y_a	decision to invest in shelter option $s \in S$
y_c	decision to invest is contraflow option $c \in C$

Second–stage Decision Variables	
f^o_k	flow on path $k \in K_o$ emanating from node $o \in O$

Table 3.1: Notation

3.1 Upper Level

For each contraflow option c in the set of contraflow options C, a binary variable y_c is assigned. Similarly, each shelter location option s in the set of such options S is assigned the binary variable y_s. The objective of the upper level is to choose decision vector $y = \{(y_s, y_c) : s \in S, c \in C\}$ such that it minimizes the maximum expected risk

exposure for all modeled scenarios. That is,

$$\min_{y} \ \mathrm{E}_{\xi} \max_{o \in O} u_o(\xi), \tag{3.1}$$

where E_{ξ} indicates the expected value operator and $u_o(\xi)$ is the risk exposure (disutility) experienced by all users emanating from origin $o \in O$ under scenario $\xi \in \Xi$.

The total cost of all decisions regarding shelters and contraflow operation is constrained by a single budget B. For costs g_c and g_s of options c and s respectively, the budget constraint is

$$\sum_{s \in S} g_s y_s + \sum_{c \in C} g_c y_c \leq B. \tag{3.2}$$

It may be that some contraflow or shelter options are mutually exclusive. For example, there could be an option for contraflow operation on each of two overlapping segments of a particular highway, but only one or the other option may be selected. For another, it could be that a proposed shelter location is only practically accessible from a road segment that is also under consideration for contraflow operation and, therefore, inaccessible when that segment is under contraflow operation. In the event that options c and c' are mutually exclusive, a set $C_i = \{c,\, c'\}$ is added to the set $C^{\perp}$, where $i \in \{1, 2, \ldots, |C^{\perp}|\}$ and $C^{\perp}$ is the set of all sets of decision variables associated with mutually–exclusive design options. Then the constraints

$$\sum_{c \in C_i} c \leq 1 \quad \forall C_i \in C^{\perp} \tag{3.3}$$

ensure that at most one option from each set of mutually–exclusive options is chosen.

These constraints serve the purpose of reducing decision variables relative to alternative formulations. Consider that the identification of contraflow and shelter options is exogenous. Thus, constraints (3.3) could be avoided by assigning a decision variable for each allowable combination of options. However, for a problem instance with N options and M mutually–exclusive combinations of options, the number of variables needed is $|\mathcal{P}(N)| - M$, where $\mathcal{P}(N)$ is the set of all subsets of N. Depending on the problem instance and solution strategy, M constraints and N variables may be more efficient than $|\mathcal{P}(N)| - M$ variables. Here, the former is adopted.

Finally, decisions are Boolean with associated binary variables

$$y_s, y_c \in \{0, 1\} \quad \forall s \in S, \forall c \in C. \tag{3.4}$$

3.2 Lower Level

To formulate the risk exposure user equilibrium, Beckmann's formulation (Beckmann et al., 1956) of the traffic assignment problem (TAP) is adopted and the objective function is modified to reflect the choice of risk–based disutility. Specifically, the integrand of Beckmann's objective is replaced with (2.4), yielding

$$\min_f Z^L = \sum_{a \in A} \int_0^{x_a(\xi)} r_a(\omega)\, d\omega. \tag{3.5}$$

The choice of Beckmann's formulation for the lower level is possible because the travel–time function and, subsequently, the risk function are separable, monotonictally increasing and convex (Larsson and Patriksson, 1995).

The flow on all used paths emanating from each origin node must equal the demand at that node.

$$\sum_{k \in K_o} f_k^o(\xi) = q^o \quad \forall o \in O. \tag{3.6}$$

The flow on link a must be equal to the sum of the flows on all paths that contain that link.

$$x_a(\xi) = \sum_{o \in O} \delta_{a,k}^o f_k^o(\xi) \quad \forall a \in A. \tag{3.7}$$

To model the effect of investment in the various options on the network, the capacity of abstract links must be limited according to whether or not the associated option, contraflow or shelter, has been selected. To achieve this, capacity side constraints (Larsson and Patriksson, 1995) are added to the lower level. In the case of shelters, each shelter option $s \in S$ is represented with an abstract link $a \in A_2$. If shelter option s has not been chosen ($y_s = 0$), then the capacity of the associated link a must be zero. That is, $p_a(y) = 0$. However, if s has been chosen ($y_s = 1$), then its capacity is equal to that shelter's design capacity p_s or $p_a(y) = p_s$.

$$x_a(\xi) \leq p_a(y) \quad \forall a \in A. \tag{3.8}$$

Such constraints apply to all links, and the cases for sheltering in place, contraflow and uncapacitated links are as follows. For sheltering in place, the capacity of the associated abstract link connecting a source node to the supersink is set equal to demand at that node. That is, $p_a(y) = q^o$. For contraflow option c and associated subgraph reconfiguration G'_c, the capacities of all abstract links incident to subgraph access nodes N_c are

infinite when that option has been chosen and zero otherwise. For all links representing road segments ($a \in A_1$), p_a is set to infinity.

Finally, the flow on all links and all paths is non–negative.

$$x_a(\xi) \geq 0 \quad \forall a \in A. \tag{3.9}$$

$$f_k^o(\xi) \geq 0 \quad \forall k \in K_o, \forall o \in O. \tag{3.10}$$

With the above formulation, the REDP is cast as a stochastic, Stackelberg game. Moreover, it is a mixed–integer problem since the decisions at the upper level are Boolean. As such, it has a non–convex feasible region.

Chapter 4: Solution Approaches

When the lower level of a Stackelberg game is modeled as an equilibrium problem, a common approach is to construct a mathematical program with equilibrium constraints or MPEC. This section discusses that process and a few approaches to the resulting stochastic program.

4.1 MPEC Formulation

For the MPEC approach, the Karush–Kuhn–Tucker (KKT) conditions of the lower level are added as constraints to the upper level, reducing the problem to a single level.

Larsson and Patriksson (1995) have shown that the KKT of the CTAP are necessary and sufficient for optimality.

Some redundancy in the constraints as stated may be eliminated by substitution. After substituting (3.7) into (3.8) and (3.9), (3.9) are equivalent to (3.10), and the constraints on the lower level become

$$q^o - \sum_{k \in K_o} f_k^o(\xi) = 0, \quad u^o(\xi), \quad \forall o \in O, \tag{3.6}$$

$$\sum_{o \in O} \sum_{k \in K_o} \delta_{a,k}^o f_k^o(\xi) - p_a(y) \leq 0, \quad \lambda_a(\xi), \quad \forall a \in A, \tag{3.8$'$}$$

$$-f_k^o(\xi) \leq 0, \quad \lambda_k^o(\xi), \quad \forall k \in K_o, \forall o \in O, \tag{3.10}$$

where $u^o(\xi), \lambda_a(\xi)$ and $\lambda_k^o(\xi)$ are the KKT multipliers associated with constraints (3.6), (3.8$'$) and (3.10), respectively, under scenario ξ. The KKT conditions, then, are (3.6), (3.8$'$), (3.10) and

$$\lambda_a(\xi) \geq 0, \qquad \forall a \in A, \tag{4.1}$$

$$\nabla_f Z^L + \sum_{a \in A} \lambda_a(\xi) \delta_{a,k}^o \geq u^o(\xi), \quad \forall k \in K_o, \forall o \in O, \tag{4.2}$$

$$\lambda_a(\xi) \left[\sum_{o \in O} \sum_{k \in K_o} \delta_{a,k}^o f_k^o(\xi) - p_a(y) \right] = 0, \qquad \forall a \in A, \tag{4.3}$$

$$f_k^o(\xi) \left[\nabla_f Z^L - u^o(\xi) + \sum_{a \in A} \lambda_a(\xi) \delta_{a,k}^o \right] = 0, \qquad \forall k \in K_o, \forall o \in O. \tag{4.4}$$

Then, following Wang and Lo (2010) and Faturechi et al. (2013), the resulting complementarity constraints are replaced with equivalent disjoint constraints, resulting in a

stochastic, mixed–integer program. The complementarity constraints (4.3) and (4.4) are replaced, respectively, with

$$L \cdot \phi_a(\xi) + \epsilon \leq \lambda_a(\xi) \leq U \cdot [1 - \phi_a(\xi)], \quad \forall a \in A, \tag{4.3a}$$

$$L \cdot \phi_a(\xi) \leq p_a^s y_a^s - x_a(\xi) \leq U \cdot \phi_a(\xi), \quad \forall a \in A, \tag{4.3b}$$

$$\phi_a(\xi) \in \{0, 1\}, \quad \forall a \in A, \tag{4.3c}$$

$$L \cdot \phi_k^o(\xi) + \epsilon \leq f_k^o(\xi) \leq U \cdot [1 - \phi_k^o(\xi)], \quad \forall k \in K_o, \forall o \in O, \tag{4.4a}$$

$$L \cdot \phi_k^o(\xi) \leq \nabla_f Z^L - u^o(\xi) + \sum_{a \in A} \lambda_a(\xi) \delta_{a,k}^o \leq U \cdot \phi_k^o(\xi), \quad \forall k \in K_o, \forall o \in O, \tag{4.4b}$$

$$\phi_k^o(\xi) \in \{0, 1\}, \quad \forall k \in K_o, \forall o \in O. \tag{4.4c}$$

The resulting program no longer has complementarity constraints but is still non–convex due to integrality constraints. To address this, Faturechi et al. (2013) constructed a similar problem by replacing the travel time function with a piecewise, linear approximation. This caused the matrix of coefficients to be totally unimodular, obviating the need for integrality constraints and resulting in a stochastic, mixed–integer program that could be solved with the integer L-shaped method. Alternatively, linear approximations of equilibrium constraints may also be constructed by using Special Ordered Sets of Type 1 variables (SOS1 variables) (Siddiqui and Gabriel, 2013).

4.2 Progressive Hedging

Another approach to stochastic problems is the progressive hedging algorithm introduced by Rockafellar and Wets (1991). Progressive Hedging (PH) is a scenario–

decomposition method frequently employed to solve multistage stochastic problems with many, branching scenarios (Veliz et al., 2014). PH iterates over fixed–scenario subproblems and penalizes those solutions most divergent from the probability–weighted average. The algorithm terminates when all deterministic solutions have converged within a certain threshold. Rockafellar and Wets (1991) showed that PH converges to the global solution for convex problems and, if it converges, to local solutions for non–convex problems. However, convergence and a global solution are not shown in general for the method.

PH has nevertheless shown some promise for non–convex problems. Indeed, Fan and Liu (2010) have successfully applied it to a bilevel, two–stage stochastic, network–design problem with equilibrium constraints. While their work bodes well for the strategy as applied to this type of problem, scenario–decomposition strategies such as PH do not mitigate the non–convexity of the MPEC formulation. Instead, each subproblem has complementarity and, if realistic values for link performance functions are chosen, high–order constraints.

4.3 An Upper–level Enumeration and Scenario Decomposition Approach

As discussed above, two common approaches to modeling related, bilevel problems introduce computationally problematic non–convexities. In this section, a case is made that the particular characteristics of this problem, especially the equilibrium sought by the lower level, will make it practical to fully enumerate the decisions at the upper level.

For a fixed, upper–level decision y and scenario realization $\xi \in \Xi$, the solution of the lower level provides the equilibrium risk exposure to users originating from all

origins under those conditions. If the solutions for all such realizations are computed and the maximum risk exposure for each $u_o^*(\xi)$ is found, the expected value of those results is the value of the upper–level objective function for that y. The decision associated with the minimum value of such results computed for all $y \in F$, where F is the set of all feasible upper–level decisions, is the enumerated solution to the REDP. Thus, the practical feasibility of solving the REDP in this manner rests on the feasibility of solving $|F| \times |\Xi|$ instances of the lower level.

Since the decisions at the upper level are Boolean, for n possible design options, $|F| \leq 2^n$. The number of possible design options is practically limited by the regional evacuation context. That is, regional planners are likely to consider several or at most dozens of contraflow options with practical relevance to the region. The decision space is further limited by the likelihood that some of the proposed design options may potentially apply to a single highway or are otherwise mutually exclusive. The number of shelter location and capacity proposals are similarly constrained. The number of scenarios $|\Xi|$, however, is not limited by such practical considerations. Fortunately, adding scenarios increases the number of lower–level instances linearly, unlike adding design options, which potentially does so exponentially.

With the number of lower–level instances limited such as they are, the practical feasibility of an enumerative solution depends on the efficiency of solving them. This efficiency, it will be shown, may be achieved by taking advantage of two characteristics of the lower–level problem. For one, lower–level realizations are independent of each other with respect to upper–level decisions and, as such, may be solved in parallel. For the other, a fast, low–memory algorithm for solving the TAP already exists. The remainder

of this section introduces a parallel approach and details the adaptation of the existing algorithm to solve the CTAP.

4.3.1 Parallel Implementation in OpenCL

The most common approach to parallel computing involves writing software for networks of discrete central processing units (CPUs). This may be done on a single, desktop machine with a multicore CPU or, more significantly, on large clusters of special–purpose machines holding a few to a few dozen CPUs each. The former, with its relatively small complement of processing units, is commensurately limited in its ability to execute instructions in parallel, while the latter is costly due to the large investment required in hardware infrastructure and high operating costs.

In the last decade, however, GPGPU has emerged as a possible compromise between hardware cost and the number of processing units available for parallel computation. Graphics processing units (GPU) are processors designed for rapid frame generation on digital displays. Due to this primary purpose, they are optimized for highly parallel, low latency operations on frame buffers. This allows GPUs to perform calculations for many, if not all, pixels of an image simultaneously. GPGPU is an approach to parallel computing that exploits these capabilities to perform other, non-graphics related tasks. While this exposes, in effect, hundreds of processors in a single graphics card, each processor is slower and has access to less memory than a typical CPU core. Despite this trade-off, workstations commonly possess both a multicore CPU and a GPU, and both may be employed to solve various tasks simultaneously. This type of environment is

often referred to as heterogeneous computing.

The adopted enumeration approach is implemented for a heterogeneous computing environment in OpenCL. OpenCL is an open framework and language specification that allows code to be compiled and executed on any processing unit on which it has been implemented. Many manufactures, including AMD, Intel and Nvidia, provide the software libraries necessary to execute OpenCL code on their CPUs and/or GPUs. General overviews of OpenCL and its utility for scientific computing may be found in (Luebke et al., 2006) and (Stone et al., 2010), while a complete introduction and specification may be found in (Gaster et al., 2012) and (OpenCL Working Group, 2011), respectively. The task of introducing OpenCL is duplicated here only in sufficient detail to describe the proposed solution strategy.

In its most abstract conception, OpenCL executes functions called *kernels* on virtual scalar processors or *processing elements*. One or more kernels are employed together as *programs* to perform some useful computation. One or more processing elements are found and managed in each hardware *device* such as a CPU or GPU. Parallelization may be achieved in this environment by writing kernels such that one or more processing elements execute an instance of a kernel for different portions of the input data. Each such instance of a kernel is called a *work item* and groups of work items may work together and share data as a *work group*. A work item is distinguished from the others in general and within its work group by its *global ID* and *local ID*, respectively.

With these concepts in place, it is possible to describe the manner in which OpenCL is applied to solve the REDP. As described above, the enumerative solution of the REDP is the best value among $|F| \times |\Xi|$ solutions of the lower level. If infeasible solutions cannot

ξ \ y	(0,0,0)	(0,0,1)	$\cdots$	(1,1,1)
1	$W_{0,1}$	$W_{1,1}$	$\cdots$	$W_{7,1}$
2	$W_{0,2}$	$W_{1,2}$	$\cdots$	$W_{7,2}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
4	$W_{0,4}$	$W_{1,4}$	$\cdots$	$W_{7,4}$

Table 4.1: Work items and their global IDs

be *a priori* excluded, as is the case here, the number of required, lower–level solutions becomes $2^n \times |\Xi|$ or N_L. To find these solutions, a kernel is written that solves the lower–level equilibrium problem for a given upper–level decision vector and scenario realization. A work item is then launched for each of the N_L solutions required and assigned a two–dimensional index $(\upsilon, \xi) = W_{\upsilon,\xi}$ as its global ID. The first value υ is from the sequence of integers $(0, 1, \ldots, 2^n - 1)$ and is the index of possible decisions ordered lexicographically, while the second value ξ is the index of the associated scenario. Note that any bijection from the integers to the set of possible decision vectors will do. Here, lexicographical ordering is chosen to exploit the ease of mapping the binary representation of the integer indexes to the possible decisions. For example, consider a REDP with three decision options (2^3 possible decisions) and four scenarios. The work item that solves the lower–level problem for decision $y = (1, 0, 1)^{\mathrm{T}}$ under scenario $\xi = 2$ is assigned the global ID $(5, 2)$ or $W_{5,2}$. It is from this index that each work item populates the capacities and risk parameters associated with a given decision and scenario realization. This example is illustrated in Table 4.1.

4.3.2 Algorithm B

Each work item must execute a kernel on its data such that the appropriate instance of the lower level is solved, resulting in the equilibrated network under those conditions. To achieve this, the kernel must implement a suitable equilibrium–finding algorithm. That is, it must be efficient and small enough to be executed on the GPU processing elements. Such an algorithm is Algorithm B (Dial, 2006). Algorithm B (B) solves the TAP by iteratively shifting flow from the used path with the highest cost to the path with the lowest cost for all O–D pairs. Dial showed that this operation, at each iteration, always improves the entire network with respect to equilibrium. By virtue of being path based, B converges more efficiently than the Frank–Wolfe algorithm (see Frank and Wolfe, 1956). Moreover, since it does not store or enumerate paths, it avoids the high memory requirements usually associated with path–based methods. As such, it is ideal for implementation as an OpenCL kernel.

In order to quickly equilibrate flows for a network, B iterates over the origins while considering only a flow–feasible, acyclic sub–network rooted at that origin. This sub–network, called a *bush*, is a subset of the network's arcs and is populated with all and only those trips emanating from its origin. B shifts flow from costly paths to cheaper paths until the bush is at equilibrium with respect to its paths or *equilibrated*. It *improves* the bush by replacing empty links with links in the opposite direction that have negative reduced costs. When the bush is equilibrated and no further improvements are possible the bush is said to be *optimal* and B moves to the next origin. When the bushes of all origins are optimal, network–wide equilibrium has been found. This process is transliterated to

pseudocode in Algorithm 1.

```
Data: network and demands for which there exists at least one feasible flow
      configuration
Result: network at user equilibrium
for all origins do
    identify an initial bush on the network and populate it with feasible flows;
end
while network is not equilibrated do
    for all origins do
        while bush is not optimal do
            while bush is not equilibrated do
                for all destinations do
                    calculate max– and min–paths;
                    shift flow from a max–path to a min–path;
                end
            end
            improve bush;
        end
    end
end
```

Algorithm 1: Algorithm B

Key to both the efficiency and validity of B is its preservation of feasibility and network improvement with each substantial change to the network — the flow–shift and bush–improvement steps. As shown in Dial (2006), the flow–shift step owes it's efficiency to the ease of computing the min– and max–paths between any two nodes in the bush. Furthermore, Dial shows that this step necessarily improves Beckmann's objective while maintaining feasibility for the entire network. Likewise, the bush–improvement step necessarily results in objective–value improvements in the next flow–shift step and, as it involves only unused arcs, retains feasibility.

4.3.3 Adapting Algorithm B to Solve the CTAP

Since the lower level of the regional evacuation problem is a CTAP and, thus, the constraints include capacity side constraints, Dial's Algorithm B is not directly applicable without modification. The problem associated with this difference in equilibrium principles arises from the possibility that a shelter link that is already operating at maximum capacity could be considered by the algorithm to be part of a minimum path. Or, stated another way, whereas B shifts flow without regard to the maximum capacity of the links, the proposed algorithm must assure that maximum capacities are not exceeded. To achieve the required modification, the unsaturated minimum path is introduced.

Definition 4 (unsaturated minimum path)**.** *A path is an unsaturated minimum path if all less costly paths contain a saturated link. That is, for origin $o \in O$, paths k and $\underline{k} \in K_o$ and associated path costs r_k^o and $r_{\underline{k}}^o$, k is an unsaturated minimum path if*

$$r_{\underline{k}}^o < r_k^o \implies \exists a \in \underline{k} \text{ such that } p_a = f_{\underline{k}}^o. \tag{4.5}$$

The modified B (B-CTAP) proceeds as B with the exception that saturated links are not considered when calculating the minimum paths. Note, however, there is no change to the calculation or use of the maximum paths. In the event that iterative improvements find a shelter as part of the maximum path, flow will be shifted to the minimum path as in any other iteration. That is, if at some state of execution, the maximum path from an origin includes a capacitated link (saturated or not) the flow–shift procedure will shift some or all flow from that path to the minimum path. This ensures that trips are never ultimately

assigned to full shelters or to a path that is more costly than using a non-full shelter. The modified algorithm is transliterated to pseudocode in Algorithm 2.

```
Data: connected network for which there exists at least one feasible flow
      configuration
Result: network at capacitated user equilibrium
identify an initial bush on the network and populate it with feasible flows;
while bush is not optimal do
    while bush is not equilibrated do
        for all destinations do
            calculate max–paths and unsaturated min–paths;
            shift flow from a max–path to an unsaturated min–path;
        end
    end
    improve bush;
end
```

Algorithm 2: Algorithm B modified for capacity side constraints (B-CTAP)

It must be asked, then, whether B–CTAP retains the properties of B which guarantee convergence on the appropriate equilibrium. As shown in Dial (2006), the flow–shift and bush improvement steps necessarily improve flows with respect to network–wide, Wardrop equilibrium. It must be shown that these steps in B–CTAP improve the capacitated equilibrium. Since B-CTAP retains the bush improvement step without modification, and the step only affects empty links with negative reduced costs, it may be accepted without further scrutiny. However, since the flow–shift step in B–CTAP ignores saturated links when searching for minimum paths, it will require some comment.

Consider a network at capacitated user equilibrium. All used paths that do not contain a saturated link have the same cost, while paths that do contain a saturated link have a cost lower than that of the unsaturated paths (Larsson and Patriksson, 1995). Under these conditions, the flows on the unsaturated paths, with respect to themselves, constitute

a Wardrop equilibrium.

By ignoring paths that contain a saturated link, B–CTAP is effectively considering a new bush containing only those links and flows present in the unsaturated paths. From the definition, it may be seen that unsaturated minimum paths are precisely minimum paths with respect to this new bush. B–CTAP then, at every flow–shift step, improves local Wardrop equilibrium for some bush in the network.

Chapter 5: Illustrative Example

To illustrate the solution strategy in a heterogeneous computation environment, a numerical example was constructed and solved using a desktop computer with a quad–core, AMD A8–5600K CPU at 3 GHz, 8 GB RAM, and an AMD Radeon HD7770 GPU. The necessary, software tool chain consisted of a compiler, in this case Microsoft Visual Studio 2013 Update 2, and the AMD APP SDK v2.9, a collection of libraries and drivers that implement OpenCL for AMD processors. Any compiler that can compile ANSI C will do, while the choice of SDK must match the processor manufacturer.

5.1 Design

The test network consists of a 3×3 grid network with two (2) contraflow options, four (4) shelters and six (6) exits. When prepared as described in Chapter 2, the network comprises 14 nodes and 53 arcs. Together, the contraflow and shelter options represent six

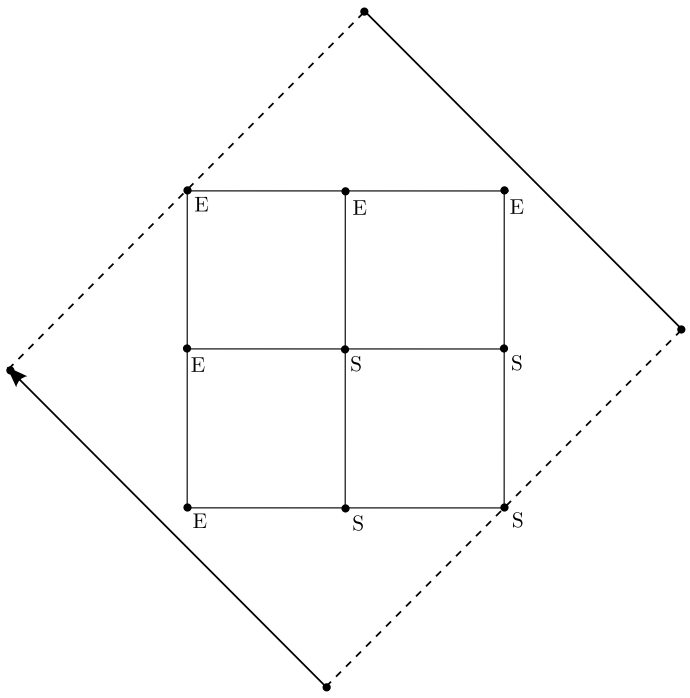

Figure 5.1: The example evacuation-zone network augmented with one contraflow option and four shelters. Nodes labeled (E) are zone exits, while (S) indicates a potential shelter site.

(6) individual options, two of which are mutually exclusive (i.e. the contraflow and do–nothing options). As shown in the figure, the segment identified for possible contraflow operation extends from opposite corners of the network. The considered option models full contraflow operation, resulting in double the capacity of the status quo. The costs and budget were chosen such that any three (3) options are feasible with respect to the budget constraint. The example network is illustrated in Figure 5.1.

Two (2) risk epicenters are modeled, both parallel to opposite extents of the network. The risk parameters were calculated for four (4) equally–likely, abstract hazard scenarios. For each risk epicenter, two (2) scenarios are modeled. One was chosen to capture an event, such as a hurricane, where high wind risk favors sheltering as a mitigation strategy except near the epicenter, where storm surge forces evacuation to shelter or

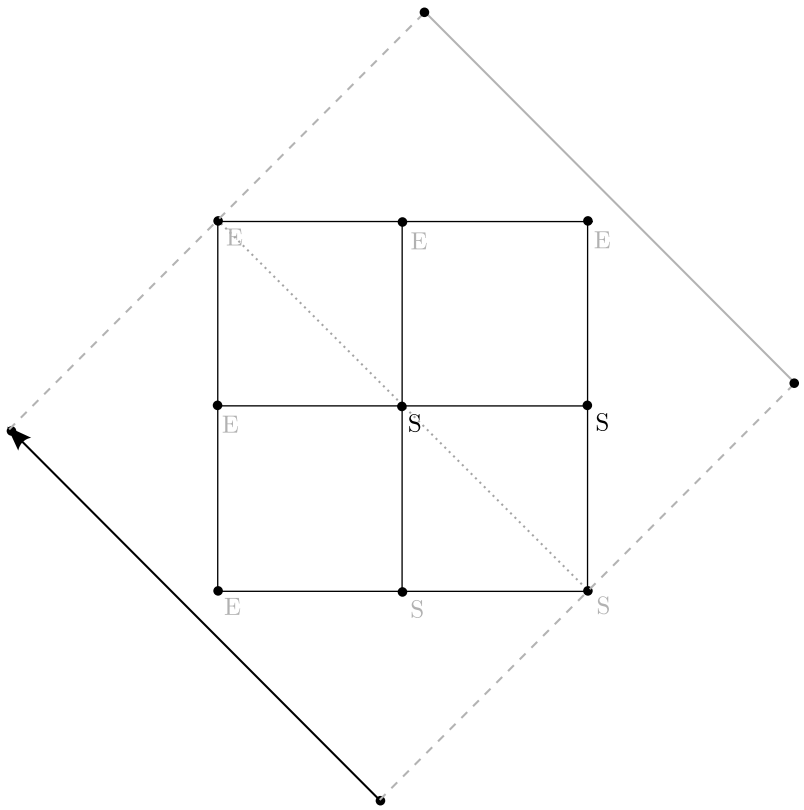

Figure 5.2: The example evacuation–zone network with the optimal solution. (S) represents a chosen shelter and the bold arrow indicates the chosen contraflow option.

zone exits. The other was chosen to model an event where a speedy evacuation to exits is favorable and sheltering is a last resort. This represents conditions associated with events such as wildfires, where sheltering provides little protection and evacuation is the primary means to reduce exposure.

5.2 Results

After execution, the optimal solution is as illustrated in Figure 5.2. Specifically, the optimal solution comprised two shelters and the decision to operate the considered road segment as a contraflow route. From the complete output of solutions it was found that the worst feasible solution was approximately three times worse than the optimal solution. It

is also the case that among the decisions that comprised two shelters only, the best among those did not choose the two shelters in the optimal solution. This further illustrates the need to consider both contraflow and shelter together in that, even if planners decided exogenously to choose contraflow and two shelters, solving the problem separately does not achieve the optimal solution.

During execution, optimality of the bushes resulting from each work item was verified by confirming feasibility and disutility equilibrium at the nodes. Here, as in Dial (2006), equilibrium at the nodes is defined as the state of affairs where the differences between the minimum and maximum disutility from each node to the sink node are within an equilibrium threshold, in this case, 0.01 trips. As such the resulting solution to the example is optimal.

Although the speed of execution was not the focus of this example, for comparison, the problem was solved in serial, in parallel on four CPU cores and in parallel on the GPU. The execution times for the above example were 200 ms, 47 ms and 137 ms respectively. With the problem scaled to the equivalent of 15 individual design options, a five–hundred–fold increase in complexity, the times were 99.1 s, 16.6 s and 8.8 s, respectively. While further scaling is the subject of continuing research, initial results suggest that the GPU

can solve a 16–option problem in less than 10 s.

Chapter 6: Future Work

With the basic framework and proof of concept established, a few avenues for future work are apparent. First, more work must be done to identify and improve the scale of problem this approach may feasibly address. Initial attempts to scale the network size have been met with process failures that suggest that execution on GPUs may be limited by memory constraints. Despite the relatively low memory requirements of Algorithm B, further memory–optimization is necessary to determine the ultimate scale of this particular solution strategy. It should also be possible, with only minor refactoring, to execute the kernel in parallel on multiple CPUs, obviating the need for extreme memory optimization at the cost of parallel execution.

Second, it may be useful to consider operational decisions after an event has been realized, such as the choice to open prepared shelters or contraflow segments. Such decisions could be modeled as upper–level recourse in the second stage. The decisions were not considered here for two reasons. One, the variety of goals and real–time information considered by emergency managers after an evacuation has begun makes it difficult to assume an objective for those decisions. That is, while minimizing maximum risk exposure may be a suitable ideal for planning for disaster, post–event circumstances, such as personnel shortages, communications failures and unexpected user behavior, add com-

plexities that change the cost and benefits of upper–level, second–stage decisions. The other is that post–event decision making is often distributed across multiple jurisdictions, requiring negotiations that are difficult to model.

However, with these considerations elaborated in sufficient detail, upper–level recourse in the second stage could be incorporated into the proposed solution strategy. Taking the simplest case as an example, planners could take, as a second stage option, the decision to enact all or none of the service offerings planned during evacuation. This single decision could be modeled in the proposed strategy by adding a third dimension to the array of work items enumerating the solution. In this case, each work item would be responsible for calculating the maximum risk exposure for a given first–stage decision, scenario and second stage decisions. While it is unlikely that costs could arise that would make the do–nothing second stage decision advantageous, the example illustrates how, in general, second–stage decision could be considered.

Finally, other decomposition strategies or meta–heuristics that extend the particular advantages of GPGPU computing should be explored for the REDP and related problems. Other user behaviors, such as dynamic or stochastic user equilibriums could be modeled in the lower level. In each case, the solution strategy proposed here could provide a valuable benchmark against efficiency and optimality.

Bibliography

Apivatanagul, P., Davidson, R. A., and Nozick, L. K. (2012). Bi-level optimization for risk-based regional hurricane evacuation planning. *Natural hazards*, 60(2):567–588.

Beckmann, M., McGuire, C., and Winsten, C. B. (1956). Studies in the economics of transportation. Technical report.

Chiu, Y.-C., Zheng, H., Villalobos, J. A., Peacock, W., and Henk, R. (2008). Evaluating regional contra-flow and phased evacuation strategies for texas using a large-scale dynamic traffic simulation and assignment approach. *Journal of Homeland Security and Emergency Management*, 5(1).

Dial, R. B. (2006). A path-based user-equilibrium traffic assignment algorithm that obviates path storage and enumeration. *Transportation Research Part B: Methodological*, 40(10):917–936.

Fan, Y. and Liu, C. (2010). Solving stochastic transportation network protection problems using the progressive hedging-based method. *Networks and Spatial Economics*, 10(2):193–208.

Faturechi, R., Isaac, S., Miller-Hooks, E., and Feng, L. (2013). Stochastic models for emergency shelter and exit design in buildings under system optimum and user equilibrium conditions.

Frank, M. and Wolfe, P. (1956). An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110.

Gaster, B., Howes, L., Kaeli, D. R., Mistry, P., and Schaa, D. (2012). *Heterogeneous Computing with OpenCL: Revised OpenCL 1*. Newnes.

Kim, S., Shekhar, S., and Min, M. (2008). Contraflow transportation network reconfiguration for evacuation route planning. *Knowledge and Data Engineering, IEEE Transactions on*, 20(8):1115–1129.

Kongsomsaksakul, S., Yang, C., and Chen, A. (2005). Shelter location-allocation model for flood evacuation planning. *Journal of the Eastern Asia Society for Transportation Studies*, 6(1):4237–4252.

Kulshrestha, A., Wu, D., Lou, Y., and Yin, Y. (2011). Robust shelter locations for evacuation planning with demand uncertainty. *Journal of Transportation Safety & Security*, 3(4):272–288.

Larsson, T. and Patriksson, M. (1995). An augmented lagrangean dual algorithm for link capacity side constrained traffic assignment problems. *Transportation Research Part B: Methodological*, 29(6):433–455.

Li, A. C., Nozick, L., Xu, N., and Davidson, R. (2012). Shelter location and transportation planning under hurricane conditions. *Transportation Research Part E: Logistics and Transportation Review*, 48(4):715–729.

Li, A. C., Xu, N., Nozick, L., and Davidson, R. (2011). Bilevel optimization for integrated shelter location analysis and transportation planning for hurricane events. *Journal of Infrastructure Systems*, 17(4):184–192.

Liu, Y., Lai, X., and Chang, G.-L. (2006). Two-level integrated optimization system for planning of emergency evacuation. *Journal of transportation Engineering*, 132(10):800–807.

Luebke, D., Harris, M., Govindaraju, N., Lefohn, A., Houston, M., Owens, J., Segal, M., Papakipos, M., and Buck, I. (2006). Gpgpu: general-purpose computation on graphics hardware. In *Proceedings of the 2006 ACM/IEEE conference on Supercomputing*, page 208. ACM.

National Weather Service (2014). Sea, lake, and overland surges from hurricanes (SLOSH). Retrieved 25 June 2014.

Ng, M., Park, J., and Waller, S. T. (2010). A hybrid bilevel model for the optimal shelter assignment in emergency evacuations. *Computer-Aided Civil and Infrastructure Engineering*, 25(8):547–556.

OpenCL Working Group (2011). The OpenCL specification, version 1.2, revision 19.

Rockafellar, R. T. and Wets, R. J.-B. (1991). Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of operations research*, 16(1):119–147.

Sherali, H. D., Carter, T. B., and Hobeika, A. G. (1991). A location-allocation model and algorithm for evacuation planning under hurricane/flood conditions. *Transportation Research Part B: Methodological*, 25(6):439–452.

Siddiqui, S. and Gabriel, S. A. (2013). An sos1-based approach for solving mpecs with a natural gas market application. *Networks and Spatial Economics*, 13(2):205–227.

Stone, J. E., Gohara, D., and Shi, G. (2010). Opencl: A parallel programming standard for heterogeneous computing systems. *Computing in science & engineering*, 12(3):66.

Urbina, E. and Wolshon, B. (2003). National review of hurricane evacuation plans and policies: a comparison and contrast of state practices. *Transportation research part A: policy and practice*, 37(3):257–275.

Veliz, F. B., Watson, J.-P., Weintraub, A., Wets, R. J.-B., and Woodruff, D. L. (2014). Stochastic optimization models in forest planning: a progressive hedging solution approach. *Annals of Operations Research*, pages 1–16.

Wang, D. Z. and Lo, H. K. (2010). Global optimum of the linearized network design problem with equilibrium flows. *Transportation Research Part B: Methodological*, 44(4):482–492.

Wolshon, B. (2001). One-way-out: Contraflow freeway operation for hurricane evacuation. *Natural Hazards Review*, 2(3):105–112.

Yuan, F. and Han, L. D. (2010). A multi-objective optimization approach for evacuation planning. *Procedia Engineering*, 3:217–227.