OPTIMIZATION AND MEASUREMENT IN HUMANITARIAN OPERATIONS: ADDRESSING PRACTICAL NEEDS

A Thesis
Presented to
The Academic Faculty

by

Mallory Soldner

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Industrial and Systems Engineering

Georgia Institute of Technology
August 2014

OPTIMIZATION AND MEASUREMENT IN HUMANITARIAN OPERATIONS: ADDRESSING PRACTICAL NEEDS

Approved by:

Dr. Özlem Ergun, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Julie Swann, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Alan Erera
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Jarrod Goentzel
Center for Transportation and Logistics,
School of Engineering
Massachusetts Institute of Technology

Dr. Joel Sokol
School of Industrial and Systems Engineering
Georgia Institute of Technology

Date Approved: 25 June 2014

To my grandparents, Marjorie & John and Harriet & Fred

ACKNOWLEDGEMENTS

Simply put, I would never have made it to this point without the support of so many people in my life. I am grateful to everyone who has helped me along the way, more than can be expressed here.

First and foremost, I would like to thank my advisors, Ozlem and Julie, who guided me at every step of the PhD. They steered me in interesting and productive research directions, challenged me to think independently and rigorously, encouraged me when things seemed bleak, and gave me the space and opportunity to actively collaborate with practitioners in my field of study. I am especially grateful for their support in these last months of the thesis.

I would also like to thank the other members of my thesis advisory committee, Jarrod, Alan, and Joel, for their time and feedback relating to this work, especially Jarrod for his guidance on our transportation delay research.

This thesis would not have been possible without the support of the Logistics Development Unit at the UN World Food Programme. I am thankful to Bernard for his vision in establishing the Georgia Tech - WFP collaboration and to Martin, Wolfgang, and Mirjana for their support. Working with LDU's supply chain team was a privilege and a pleasure, and I especially thank Sergio, Rui, Zmarak, and Luis. Rome became a second home to me during these years, and I am grateful for the nice times and travels shared together with Madeline, Cameron, Amin, Claudia, Edita, and other friends in Italy and now around the world.

At Georgia Tech, I am grateful to the world-class faculty that I had the privilege to learn from and to the staff of ISyE for their support, especially to Pam and Yvonne for their encouragement and help with so many of the details.

My fellow graduate students were one of the biggest highlights of my time at ISyE. I thank Tugce for being the best office-mate on the planet and being there through thick and thin, Daniel and Tim for so many nights grilling on the porch, and Veronique, Ilke, Monica,

Soheil, Bahar, Isil, Andres, Stefania, Murat, Ezgi, Diego, Rodrigue, Vinod, Norbert, and many others for their support and friendship throughout these years.

Finally, I would like to thank my parents and family. I am thankful to my Mom for always supporting me, to Ron for always being a phone call away, and to my Dad for his encouragement. I also recognize Brigid, Gretchen, and Rachel for being lifelong friends, and my Godson, Nicholas, and niece, Elizabeth, for always making me smile.

Contents

List of Tables

List of Figures

SUMMARY

This thesis focuses on three topics relevant to humanitarian applications: (i) stable and complete assignment of staff members to field offices, (ii) bottleneck management for transportation networks, and (iii) performance measurement of the food assistance supply chain.

The assignment and reassignment of personnel to jobs is a large-scale problem faced by many organizations including the military and multi-national organizations. Although successful algorithms have been developed that can ensure matchings that are stable (without incentive to deviate), not all practical concerns have been addressed by these algorithms. For example, the gap we study is that when staff members do not provide preference lists covering all jobs, a complete stable matching is not guaranteed. In the first part of the thesis, we model negotiations, which occur in practice, as part of the problem of matching all agents. We introduce algorithms and structural results for when the organization negotiates with specific agents to modify their preference lists and the centralized objective is to minimize the number or cost of negotiations required to achieve complete stable matchings.

An uncertain environment with disruptions is a reality faced by many humanitarian operations but not fully addressed in the literature. Transportation delays are often driven by reliability issues (e.g., customs delays, strikes, and the availability of transport), and the length of wait time can be influenced by congestion. In the second part of the thesis, we describe a queuing model with breakdowns to model delays in port and transportation corridors (the overland travel from discharge ports to delivery points). Using the model, we gain insights into where delays are most detrimental to system performance (i.e., the network's "bottleneck") in port and transportation corridors. We then include our delay modeling in a convex cost network flow model that determines optimal routing when several port and corridor options are available. Finally, we examine a resource allocation model for where to invest in improvements to minimize delay. Throughout, we compare solutions using the optimal approach to rules of thumb and identify important factors that might be missing in practical decision making currently.

Third, we present a case study on the implementation of supply chain key performance indicators (KPIs) at a large humanitarian organization. We describe (i) the phases necessary for a full implementation of supply chain KPIs at a humanitarian or non-profit organization, (ii) how to address strategy, mindset, and organizational barriers, and (iii) how to adapt commercial supply chain KPI frameworks to the humanitarian sector, factoring in implementation constraints present in the humanitarian sector that may impact KPI development.

Last, a conclusion chapter discusses areas where this research may or may not generalize for each of the three topics studied.

Chapter I

INTRODUCTION

In this thesis, operations research and management science techniques are applied to practical humanitarian topics, namely stable and complete assignment of staff members to field offices, bottleneck management for transportation networks, and performance measurement of the food assistance supply chain. In each area, the work addresses a specific practical need: stable assignments where all agents are matched, decision support for humanitarian transport planning that includes congestion and disruptions in the system and does not assume deterministic data inputs, and implementation of performance measurement in the humanitarian supply chain.

This work is motivated through a close collaboration with the Logistics Development Unit of the United Nations World Food Programme (WFP). WFP is the United Nations' frontline agency mandated to combat global hunger. The collaborative research was enabled by over 15 months of work experience at the institution in addition to 2.5 years of remote consulting by the author. Novel modeling techniques are used to address the characteristics of the humanitarian applications studied, though the research may be applicable in other contexts as well. Insights are generated from investigating the problem structures and the new algorithms developed to solve these models. The next subsections describe the respective thesis chapters devoted to each of the topics investigated.

1.1 Chapter 2: Stable Assignment Problems for Staffing: Negotiated Complete Stable Matchings

Large-scale assignment (and reassignment) of staff to jobs is important in many industries including ones such as graduates from medical school to residency or military assignment of officers. The US Navy alone reassigns 300,000 personnel to jobs annually [58] and the National Resident Matching Program (NRMP) assigns 30,000 medical school graduates in the US each year [73]. Although successful algorithms have been developed that can ensure matchings that are stable (without incentive to deviate), not all practical concerns have been addressed by these algorithms. For example, the gap we study is that when staff members do not provide preference lists covering all jobs, a complete stable matching is not guaranteed.

Our objective is to identify mechanisms to find "complete matches" where all staff and jobs are assigned without submitting full preference lists. In our research on complete matches, we model an organization's negotiations with staff or jobs on their preferences to find complete and stable matchings. Staff preferences can be influence by promotions, titles, or promises for better future assignments. Job preferences can be influenced through staff member training or incentives offered to hiring managers.

More specifically, we model negotiations, which occur in practice, as part of the problem of matching all agents. We provide mathematical programming formulations that result in negotiated complete stable matchings, minimizing the number or cost of negotiations, in which (i) all staff and jobs are matched and (ii) no blocking pairs exist according to the negotiated preference lists. Two negotiation schemes*, Append-to-End* and *Extend-Thru*, are investigated, each having differing stability requirements and differing assumptions on how preferences are added to the end of an agents' preference lists through negotiation.

When the centralized objective is to minimize the number of negotiations required, we show that under *Append-to-End*, $N - M$ negotiations are required. Here, N is the problem size and M is the number of pairs matched in a stable matching for the instance prior to negotiations, and we also develop a polynomial-time *Naive Algorithm* that achieves an optimal solution. Under *Extend-Thru*, we show that the problem can be solved optimally through linear programming. Compared to the existing literature on 'almost stable' matches, we find that our 'almost acceptable' mechanisms may require significantly fewer compromises to reach a complete matching (e.g., only 37 negotiated pairings instead of 400 blocking pairs in matching 782 medical students to hospitals).

When the centralized objective is to minimize the cost of negotiations, we introduce a generic cost minimization objective function and provide specific analysis for the *Count Agent Negotiations Cost Function,* which minimizes the number of negotiations with specific agents. Under *Extend-Thru*, and as generalizes for any linear cost objective for this negotiation scheme, we show that the problem can be solved in polynomial time. Under *Append-to-End*, we find that a linear relaxation of the IP is not guaranteed to produce an optimal solution, and we introduce a heuristic with fast solution times, even for very large problems, that is simple to implement.

Last, we extend our concept of negotiation modeling to include negotiating for changes to the sequencing within staff and job rankings. We introduce the negotiation scheme, *Move-to-Beginning,* which assumes that negotiation incentives are strong enough so that whenever a pair is selected for negotiation, the staff member and job in the pair both move each other to the front of their respective preference lists. We introduce an IP for minimizing the number of negotiations required under *Move-to-Beginning.* As in *Append-to-End*, we show that again surprisingly $N - M$ negotiated pairs are required under *Move-to-Beginning* to create a negotiated complete stable matching (which can be found through a variant of the *Naive Algorithm*). This is true despite *Move-to-Beginning* being a stronger negotiation tactic than *Append-to-End* in that more blocking pairs are eliminated per negotiation.

This research contributes by studying optimization with decentralized decision makers in the context of developing algorithmic approaches to large-scale and practical stable matching problems. Currently, it is not possible to guarantee complete matches without preferences over all jobs and people. We develop innovative approaches to resolve this problem by negotiating preferences when preference lists are truncated.

1.2 Chapter 3: Managing Bottlenecks in Port and Overland Transport Networks for Humanitarian Aid

Delays in humanitarian supply chains prevent life-saving aid from reaching beneficiaries when needed. Characterizing and reducing these delays in the transportation network is the focus of Chapter 3. The gap studied is how to provide routing and network improvement decision support for humanitarian networks, incorporating congestion and disruptions and without requiring detailed data on costs and capacity over time.

In humanitarian applications, transportation delays are often driven by reliability issues (e.g., customs delays, strikes, and the availability of transport), and the length of wait time can be influenced by congestion. We first introduce a queuing-based model with stochastic server breakdowns for quantifying congestion in ports and corridors with closed-form expressions for expected waiting time in the system. We show that our delay function is convex with respect to flow, introduce a convex cost flow model for routing flow to minimize delay, and give optimality conditions for minimizing port and corridor congestion delays for a structured network type. Finally, we characterize the monotonic impact of parametric changes on total wait, and we formulate a mathematical program that simultaneously invests a budget and routes flow optimally through the network.

Throughout, we compare solutions using the optimal approach to rules of thumb and identify important factors that might be missing in practical decision making currently. Overall, we create models that do not require precise or extensive inputs and for which most of the realistic-sized instances evaluated can be solved quickly and with open-source optimization.

1.3 Chapter 4: A Case Study on Implementation of Supply Chain Key Performance Indicators at a Large Humanitarian Organization

In the last chapter, we study a large-scale implementation of key performance indicators (KPIs) in the context of the humanitarian supply chain in order to document the complex process and impact future implementations in humanitarian organizations. The research approach is a single-case study on the implementation of supply chain key performance indicators at a large humanitarian organization, using action research methodology.

Our action research case, addressing performance measurement in the humanitarian supply chain, has a scope not shared by any existing papers in the literature to the best of our knowledge. We describe (i) the phases necessary for a full implementation of supply chain KPIs at a humanitarian or non-profit organization, (ii) how to address strategy, mindset, and organizational barriers, and (iii) how to adapt commercial supply chain KPI frameworks to the humanitarian sector, factoring in implementation constraints present in the humanitarian sector that may impact KPI development.

As a single case study, the findings may not generalize. Further cases and research are recommended to better characterize performance management implementations in the humanitarian context. This case study can be used as reference for a humanitarian or non-profit agency undertaking or considering a supply chain performance measurement initiative.

1.4 Chapter 5: Conclusion

Chapter 5 concludes the thesis with a discussion of areas where this research may or may not generalize. While all three areas in this dissertation were motivated through observations of real humanitarian operations and were tailored to fit the humanitarian context, we note that the research may be applicable in other contexts as well.

First, our negotiated complete stable matching models can generalize to fit many existing applications of stable matching where agents provide truncated preference lists (e.g., in assigning military personnel to posts or in matching medical residents to hospitals, where it is unrealistic for a staff members to rank all possible posts due to the large-scale nature of the assignments). Second, inherent uncertainty exists in transportation networks in many settings and contexts, and the developed congestion, routing, and investment models may generalize to other public health and private sector applications, though we note cases where the underlying assumptions of the models may not be the right fit. Third, we discuss ways in which insights from the case study on the implementation of supply chain KPIs at a large humanitarian organization may apply to other non-profit and for-profit organizations depending on corresponding fit to the case in terms of the centralization of information and decision-making and the availability of downstream data.

Chapter II

STABLE ASSIGNMENT PROBLEMS FOR STAFFING: NEGOTIATED COMPLETE STABLE MATCHINGS

2.1 Introduction

Large-scale assignment (and reassignment) of staff to jobs is important in many industries including ones such as graduates from medical school to residency or military assignment of officers. The US Navy alone reassigns 300,000 personnel to jobs annually [58] and the National Resident Matching Program (NRMP) assigns 30,000 medical school graduates in the US each year [73]. Several algorithms [72] have been developed to assist in this process, where they may focus on ensuring matching solutions with stability (i.e., no individual and job prefer each other to their assignment). The importance of work in this area was recently recognized with a Nobel Prize in Economics for some of the researchers [57]. The idea behind many of the algorithms is simple: staff have a list of ordered preferences for jobs (and jobs have preferences for staff); iterations are made for each staff member, where each in turn "proposes" to the next job on their list that is unassigned; if the proposed job prefers that staff to their current assignment then a new match is made and the other personnel is returned to the unmatched set. The elegance of the algorithm is that the matches that arise from the staff-first algorithm (or job-first, if jobs are cycled through instead) are (i) guaranteed to be a solution where no person or job will want to switch with another, (ii) solved in reasonable time, and (iii) are the staff-optimal solution (or job-optimal, respectively). This combination of economic concepts in an algorithmic framework also has importance in other areas such as the housing market [80], systems with school choice [1], or matches for kidney exchanges [75].

However, there are many practical considerations that have not yet been fully addressed in the research. In particular, *our objective is to identify mechanisms to find "complete matches" where all staff and jobs are assigned without submitting full preference lists.* In our research on complete matches, we model and organization's negotiations with staff (or jobs) regarding their preferences to enable complete matchings. Each of these topics is important for a variety of organizations, but we have identified them through our close collaboration with the United Nations World Food Programme (WFP). WFP alone reassigns over 500 people each year with each reassignment costing on average $40,000 per person [59]. The process can take WFP's human resources department over three months to finalize with many negotiations happening throughout to make sure that all jobs and personnel get matched. In order to fill the growing number of hardship duty stations, negotiations can involve promises for future assignments, promotions, and career advancement strategies. Stable matches can lead to improved stability of the workforce, reduced costs of assignments, and ultimately more dollars available for beneficiaries. We develop mathematical models, investigate the complexity and structure of the underlying problems, develop algorithms to find solutions, and analyze the algorithms for their performance in terms of running time or solution quality as the problem size grows.

This chapter is organized as follows. Section 2.2 provides foundational stable matching results and notation, highlights the contributions of this research, and summarizes related literature. In Section 2.3 we describe our approach and results, introducing negotiation mechanisms and providing structural and algorithmic results for when the centralized objective is either to minimize the number or cost of negotiations. We conclude in Section 2.4 with a summary of the work and directions for future research.

2.2 Literature Review

2.2.1 Stable Matching Background and Concepts

Succinctly stated, the *stable matching problem* seeks to pair *agents* on two sides of a bipartite graph into matchings that are *stable.* In our work, the agents on each side of the bipartite graph are staff and jobs, and we will assume an equal number in each set. *Preferences* are expressed by each staff and job through a ranked list of agents on the opposite side of the graph.

When all of the staff or jobs are listed, a preference list is called *complete*, and when

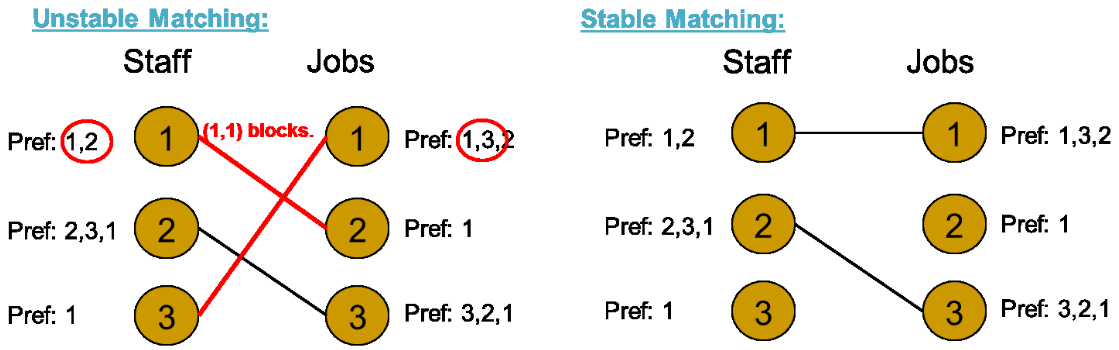


Figure 1: Matching Instance: Unstable Matching (Left) and Stable Matching (Right)

only a subset is listed, a preference list is called *truncated*. A pair (i, j) is *acceptable* if and only if i appears on j's preference list and j appears on i's preference list.

A *complete matching* is one in which all staff and jobs are paired. A matching is *stable* when there is no staff and job pair who prefer each other to their current match; such a pair would be called *blocking*. By convention, we assume than an agent prefers being matched to being unmatched. Figure 1 illustrates a matching problem with three agents of each type, with the preferences indicated. The matching indicated by arcs on the left is *unstable* because the staff, job pair (1,1) *blocks* (since each prefers the other over their current match (2 and 3, respectively). The matching on the right is stable with no blocking pairs although it is not complete.

If all agents submit complete preference lists, then complete stable matchings can always be found [35]. However, *when some agents submit truncated preference lists, the stable matchings can leave staff and jobs unpaired* [36]. We address this limitation in our research.

2.2.2 Related Literature and Contributions

Seminal work by Gale and Shapely introduced stable matching and showed that every problem instance has a stable matching that can be found in polynomial time through a deferred acceptance algorithm [35]. In their algorithm, the staff-optimal stable matching is found through rounds of staff proposals to their most preferred jobs until no further proposals are possible. Acceptance of staff proposals from the jobs is temporary until staff proposals are exhausted, with jobs able to leave temporary proposals for better offers. It is known that

Table 1: Summary of Notation

Notation	Description
N	problem size (the number of staff members and the number of jobs)
M	size of the stable matching before negotiations (# of pairs chosen)
I	the set of staff members
J	the set of jobs
A	the set of acceptable pairings given the preference lists (prior to any negotiations)
P_k	the ranked preference list for staff or job k (prior to any negotiations)
A_k	subset of jobs or agents in P_k that also list k in their preference lists
$k >_i j$	shorthand for $\{k : \ k >_i j\}$, the set of jobs that staff member i prefers to j
$k >_j i$	shorthand for $\{k : \ k >_j i\}$, the set of staff members that job j prefers to i
x_{ij}	the decision variable for each acceptable pairing (1 if chosen, 0 otherwise)
y_{ij}	the decision variable for each unacceptable pairing (1 if negotiated, 0 otherwise)
z_{ij}	the decision variable for each acceptable pairing (1 if negotiated, 0 otherwise) (only used in the *Move to Beginning* Negotiation Scheme)

all stable matchings in an instance (for either complete or truncated preference lists) are of the same cardinality and involve the same agents [36]. Good overviews of the area are [38, 48, 56, 72, 76], and the references therein.

Interestingly, stable matchings can also be found through linear programming [74], which we build upon in our work. See Formulation 1 for the problem formulation that is generally used [74], where the notation is summarized in Table 1. The decision variables x_{ij} correspond to whether pair (i, j) is chosen in a matching. For each acceptable arc $(i, j) \in A$, $x_{ij} = 1$ if the pair is in the solution vector or 0 otherwise. Constraints (2) and (3) ensure that no more than one staff member is assigned to a job and that no more than one job is assigned to a staff member. Constraint (4) (written for each acceptable pair), ensures that none are blocking pairs, by saying that at least one of the following holds: (a) staff i is assigned to job j, (b) staff i is assigned a job more preferred than j, or (c) job j is assigned a staff more preferred than i. If any of the latter conditions hold, then (i, j) cannot be a blocking pair to the final solution. Gale and Sotomayor [36] show that all feasible solutions have the same objective function value and Roth et al [74] prove that this linear program has an integral polyhedron, implying that integral solutions can be found in polynomial time.

Formulation 1 Linear Programming (LP) formulation for finding a maximal stable matching

$$
\begin{array}{rrll}
max & \sum_{(i,j)\in A} x_{ij} & & (1)\\
s.t. & \sum_{j\in A_i} x_{ij} \leq 1 & \forall i \in I & (2)\\
 & \sum_{i\in A_j} x_{ij} \leq 1 & \forall j \in J & (3)\\
 & x_{ij} + \sum_{(k>_i j)\in A_i} x_{ik} + \sum_{(k>_j i)\in A_j} x_{kj} \geq 1 & \forall (i,j) \in A & (4)\\
 & 0 \leq x_{ij} \leq 1 & \forall (i,j) \in A & (5)
\end{array}
$$

Many matching problems in practice require a complete matching, where keeping matches stable is also a goal. For example in the U.S. Navy Enlisted Assignment Process, the assignment of sailors to billets has been documented in [69, 90] as having a high priority to assign all sailors while also covering the most important billets. Likewise, it is preferable to assign all students to schools in school choice mechanisms [2] and to pair as many kidney donors and recipients as possible in kidney exchanges [75].

However, many of these systems are large-scale with hundreds to thousands of agents on each side and it is not reasonable to expect agents to submit complete preference lists. One stream of literature in stable matching with truncated lists focuses on truncation strategies for agents [26, 32, 77]. Another stream of literature focuses on admitting the fewest blocking pairs into a solution to maximize the size of a matching, creating so-called 'almost stable' matchings [39, 17].

Similar to our research, the 'almost stable' matching literature focuses on pairing as many agents as possible. Unlike our work, they strictly enforce the truncated preference lists submitted by agents but allow deviation from a purely stable matching to include blocking pairs. On the other hand, in our research, we do not allow blocking pairs but we deviate from the classic stable marriage problem by negotiating 'unacceptable' assignments in which at least one agent in a pairing was not listed by the other agent. Biro et al [17] show that the 'almost stable' matching problem is NP-hard, and Hamada et al [39] introduce approximation lower bounds. On the other hand, we show that our negotiation algorithms that minimize the number of negotiated preferences needed to create a complete matching can be solved in polynomial time.

Extending preference lists to include more choices for agents is also closely related to our work, as it is a means for turning unacceptable assignments into acceptable, negotiated ones. In evaluating whether the U.S. Navy could adopt a stable matching assignment process, Robards and Gates [69] note that one way to assign as many sailors to billets as possible is to force each sailor to list and rank all possible billets that he/she is eligible for (including those over- and under-qualified for and those which are not desired). In the case where all staff and job assignments are possible (even if not desired by individual agents), which is what we study, the technique described in [69] is equivalent to forcing complete preference lists. Our proposed models are novel in that they seek minimal additions or modifications to agent preferences, while maintaining that the final result meets stability properties.

2.3 Approach and Results

For matchings coordinated by a centralized organization, there are opportunities to design mechanisms that will result in complete matchings. We study negotiations between a central organization and agents, where negotiations change staff and job preferences. For example, we assume that an organization could persuade a staff to add a particular job to their preference list through job promotions, monetary bonuses, or other mechanisms. At WFP, difficult assignments are often linked to promotions and increased pay. Similarly, the organization could influence a job to add a staff to its preference list (e.g., with job training). We model several negotiation techniques used in practice as mechanisms for modifying preference lists while trying to achieve complete matchings. We call these *negotiated complete stable matchings* and discuss algorithms and structural results for when the centralized objective is to minimize the number or cost of negotiations.

2.3.1 Negotiation Mechanisms

Continuing with the example instance with an incomplete stable matching (on the right of Figure 1), we see that a central matchmaker might want to focus negotiation efforts on the unmatched staff and jobs, Staff 3 and Job 2, neither of which find the other acceptable. Figure 2 illustrates that if Staff 3 and Job 2 can be incentivized to add each other to their respective preference lists (e.g. with a promotion and training offered to Staff 3 for Job 2), then after this compromise, the resulting instance under negotiated preference lists will have a complete stable matching in {(1,1), (2,3), (3,1)}.

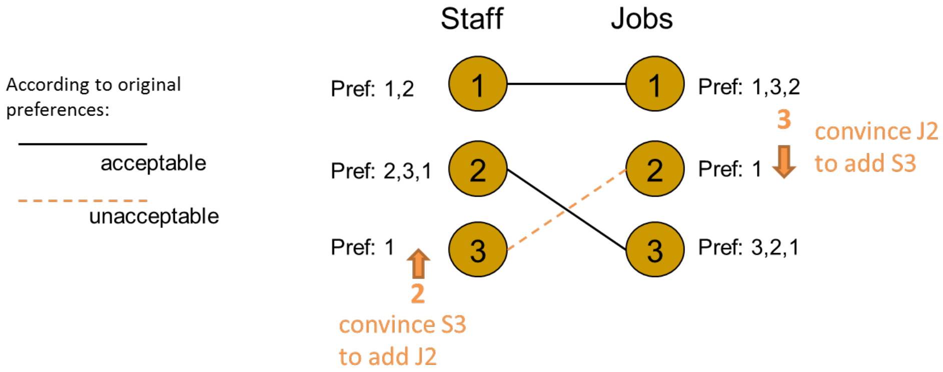


Figure 2: Negotiation can be used to achieve complete stable matchings.

We show that through negotiation schemes, such as appending preferences to the end of staff and job preference lists, complete stable matchings according to the new negotiated preferences can be found. We refer to the revised staff and job preference lists after negotiations as *negotiated preference lists.*

Our *negotiated complete stable matchings* have two key properties: (i) that *all staff and jobs are matched* and (ii) that *no blocking pairs exist according to the negotiated preference lists.* Within this definition, multiple negotiation schemes are possible. For example, consider the negotiation scheme where previously unranked staff and jobs are appended to the end of staff and job preference lists. As staff and jobs are appended to preference lists, the number of admissible pairings increases (by the inclusion of pairs made acceptable through negotiation such as (3,2) in Figure 2), making it easier to achieve an almost acceptable complete, stable matching. We call this negotiation scheme *Append-to-End.* On the other hand, in our other mechanism investigated, *Extend-Thru*, multiple agents are appended to the end of a preference list in a ranked order. We present results for *Append-to-End* in Section 2.3.2.1 and for *Extend-Thru* in Section 2.3.2.2.

Since complete preference lists always result in the existence of complete, stable matchings [35], almost acceptable complete stable matchings must also always exist under the negotiation scheme where every unlisted staff and job is appended to each staff and job's preference list (until each staff and job has a complete preference list). Such a naive negotiation strategy defeats the point of truncated preference lists and would likely be difficult for cases with many agents and costly if monetary incentives are offered for each negotiation. Thus a natural research question that arises is *what minimal set of negotiations can achieve 'almost acceptable' complete stable matchings?*

2.3.2 Minimizing the Number of Negotiations

We show that mathematical programming can be used to create almost acceptable complete stable matchings for various negotiation schemes, and we introduce an integer program, called *minNegotiations,* in Formulation 2 to find a minimal set of negotiations to achieve an almost acceptable complete stable matching for a problem instance. This is a baseline formulation, leaving a generic placeholder for the stability constraint, since stability conditions vary depending on the negotiation scheme considered (due to different impacts on staff and job preference lists). For example, depending on whether negotiations modify a preference list by one or multiple choices, the blocking pair prevention expression varies.

The objective, which is given in Expression (6), is to minimize the number of unacceptable pairings introduced in the matching, each one representing a negotiation. Defined only for acceptable pairs,$(i, j) \in A$, x_{ij} is a binary variable that is 1 if staff member i is assigned to acceptable job j and 0 otherwise. Defined only for unacceptable pairs, $(i, j) \in (I \times J)\backslash A$, y_{ij} is a binary variable that is 1 if staff member i is assigned to job j (through a negotiation on (i, j)) and 0 otherwise. Constraints (7) and (8) ensure that a complete matching is found. The generic stability placeholder constraint is given by Constraint (9).

Formulation 2 Base *minNegotiations* Formulation: 'Almost Acceptable' Complete Stable Matching Models

$$
\begin{aligned}
min \quad & \textstyle\sum_{(i,j)\in I\times J\setminus A} y_{ij} & & (6)\\
s.t. \quad & \textstyle\sum_{j\in A_i} x_{ij} + \sum_{j\in J\setminus A_i} y_{ij} = 1 \quad \forall i \in I & & (7)\\
& \textstyle\sum_{i\in A_j} x_{ij} + \sum_{i\in I\setminus A_j} y_{ij} = 1 \quad \forall j \in J & & (8)\\
& \text{stability constraint(s)} & & (9)\\
& x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A & & (10)\\
& y_{ij} \in \{0,1\} \qquad \forall (i,j) \in (I\times J)\setminus A & & (11)
\end{aligned}
$$

For both negotiation mechanisms investigated, *Append-to-End* and *Extend-Thru*, which are fully defined in the next two sub-sections, we show that solving *minNegotiations* will always produce a feasible, almost acceptable complete stable matching, since its polyhedron is non-empty (Theorem 1).

Theorem 1. *There is a non-empty feasible region for minNegotiations under negotiation schemes Append-to-End and Extend-Thru (defined in Section 2.3.2.2).*

Proof. The reader is referred to Proof Appendix A.1.2. □

2.3.2.1 Negotiation Scheme: Append-to-End

An important assumption for *Append-to-End* is that for each staff or job k, k is indifferent in preference to those staff or jobs not listed in P_k (this will be relaxed in *Extend-Thru*). As a result, unacceptable pairs cannot be blocking pairs (since both the staff member and job cannot strictly prefer each other in an unacceptable blocking pair, otherwise the pair would have been acceptable due to both the staff and job member ranking each other). Thus, it must only be verified that no acceptable pairs block the solution matching. For the negotiation scheme *Append-to-End,* the corresponding stability constraint that enters Formulation 2 in place of Constraint (12) is given in Equation (12).

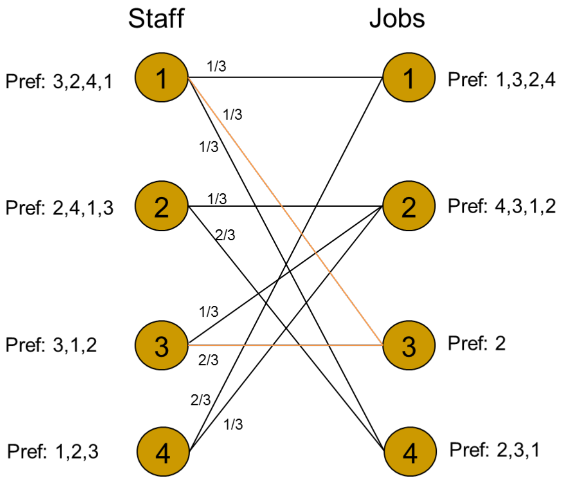


Figure 3: Fractional optimal solution to *minNegotiations* under *Append-to-End.*

$$x_{ij} + \sum_{(k >_i j) \in A_i} x_{ik} + \sum_{(k >_i j) \in P_i \setminus A_i} y_{ik} + \sum_{(k >_j i) \in A_j} x_{kj} + \sum_{(k >_j i) \in P_j \setminus A_j} y_{kj} \geq 1, \ \forall (i,j) \in A \quad (12)$$

Constraint (12) ensures that no acceptable pairs are blocking, by saying that either i is assigned to j or to a job more preferred than j or j is assigned to a staff member more preferred than i. If any of these conditions hold, then (i,j) cannot be a blocking pair in the final solution. In the constraint, the summations involving the x variables are over preferred staff and jobs that are part of acceptable arcs, while the y variables are over preferred staff and jobs that are part of unacceptable arcs prior to negotiations (e.g., for staff i, the set of jobs that i ranks in P_i but that do not in turn include i on their preference lists).

Next, we show that for *Append-to-End*, the linear relaxation of *minNegotiations* does not have an integral polyhedron (Proposition 2).

Proposition 2. *The linear relaxation of minNegotiations (under the negotiation scheme Append-to-End) does not have an integral polyhedron.*

Proof. This proof is by counterexample. Figure 10 shows a fractional, optimal vertex solution *minNegotiations* under *Append-to-End* that was found using the simplex algorithm implementation in Gurobi Optimizer 5.6 *[61].* □

The implication of Proposition 2 is that, for solving larger instances, integer programming

Algorithm 2.1 Naive Negotiation Algorithm

```
1. First, obtain a stable matching to the problem instance. Many techniques are available
   (e.g. see polynomial time algorithms in [35, 38, 74]). Let $M$ be the number of pairs
   matched.
2. Construct an $unmatchedStaff$ list of the $N - M$ staff members that are unmatched
   and an $unmatchedJobs$ list of the $N - M$ jobs that are unmatched.
3. While $unmatchedStaff$ is non-empty:
   (a) Remove some staff member $i$ from unmatchedStaff and some job $j$ from
       $unmatchedJobs$
   (b) Negotiate $(i, j)$ to be paired (for Append-to-End this means appending $i$ to the
       end of $P_j$ and/or $j$ to the end of $P_i$ if they are not already listed)
```

may not be an efficient methodology to produce feasible solutions to $minNegotiations$ for this negotiation scheme [53]. Thus, an algorithmic approach is motivated to determine the minimum number of negotiations needed for an instance.

A *Naive Algorithm* is proposed in Algorithm 2.1 that starts with a potentially incomplete stable matching and arbitrarily forces unacceptable pairings through negotiation until all staff and jobs are matched. *Naive Algorithm* begins by creating a stable matching of M pairs of the N total staff and jobs, leaving $N - M$ staff and jobs unmatched. The algorithm then negotiates to create an a pairing between each unmatched staff and one job, resulting in $N - M$ negotiated arcs. In the algorithm, Step 1 to produce the stable matching is polynomially solvable ([35, 38, 74]), and Steps 2 and 3 for negotiations are $O(N)$, making the overall solution time of *Naive Algorithm* polynomial.

In Step 3(b) of the Naive Algorithm, we further clarify that each $(i, j) \notin A$ (Lemma 3). The implication is that *Append-to-End* can indeed create a negotiated pairing by appending i to the end of P_j and/or j to the end of P_i, for the one or both of the agents does not already list the other.

Lemma 3. *Consider a stable matching x. Let staff i and job j be unmatched in x. Then, $(i, j) \notin A$.*

Proof. For a given stable matching instance, let x be a stable matching, and let staff i and job j be unmatched in x. Assume for the sake of contradiction that $(i, j) \in A$. Since

i and j are unmatched in x, $\sum_{k \in A_i} x_{ik} = 0$ and $\sum_{k \in A_j} x_{kj} = 0$. Further, since x is a stable matching, by Constraint 4, we have $x_{ij} + \sum_{(k >_i j) \in A_i} x_{ik} + \sum_{(k >_j i) \in A_j} x_{kj} \geq 1$. Yet, $\sum_{k \in A_i} x_{ik} = 0$ and $\sum_{k \in A_j} x_{kj} = 0$ imply that $x_{ij} + \sum_{(k >_i j) \in A_i} x_{ik} + \sum_{(k >_j i) \in A_j} x_{kj} = 0 < 1$. Here, we have reached a contradiction. □

The question, then, is whether optimal negotiations could use fewer than the $N - M$ negotiations required by *Naive Algorithm.* For the negotiation scheme *Append-to-End,* we show that, surprisingly, a solution to $minNegotiations$ with fewer than $N - M$ negotiations does not exist (Theorem 4).

Theorem 4. *The optimal solution value of the minNegotiations is $N - M$ under the Append-to-End negotiation scheme.*

Proof. Since *Naive Algorithm* results a feasible solution to $minNegotiations$ under *Append-to-End* for any problem instance, $N - M$ is an upper bound for the model. We also have that $N - M$ is also a lower bound for the solution value to $minNegotiations$ under *Append-to-End* (see Lemma 31 in Proof Appendix A.2). Thus, we can conclude that $minNegotiations$ under *Append-to-End*'s optimal solution value is $N - M$. □

Important to the proof is Lemma 31 which is based on several preliminary results that also appear in Proof Appendix A.2. For example, we give corollaries to Theorem 1.4.3 in [38] which are useful in describing the impact of appending a single preference to the end of a staff or job preference list (Corollaries 26 and 27). Also Theorem 29 bounds the increase in the cardinality of pairs matched as a result of any single negotiated pairing (involving up to two simultaneous preferences being appended to the end of preference lists). A corollary to Theorem 4 is then that *Naive Algorithm* produces an optimal solution to $minNegotiations$ (Corollary 5).

Corollary 5. *The polynomial time Naive Algorithm achieves a minimum negotiation solution to minNegotiations under the Append-to-End negotiation scheme.*

Proof. Naive Algorithm produces an $N-M$ negotiation feasible solution to *minNegotiations.* From Theorem 4 we have that the optimal solution value of the *minNegotiations* is $N-M$ for *Append-to-End.* □

The prospect of only requiring $N-M$ negotiations may be quite powerful in practice. For example, in the case of the Scottish Foundation Allocation Scheme investigated in [17], 400 blocking pairs needed to be admitted into the solution to match all 782 residents under the 'almost stable' paradigm. On the other hand, only 37 negotiated pairings would be required using $minNegotiations$ and the 'almost acceptable' paradigm.

2.3.2.2 Negotiation Scheme: Extend-Thru

In *Extend-Thru,* staff and jobs are assumed to have strict preferences over unacceptable pairs, and negotiations extend a ranked unacceptable list through the preference negotiated. For example, if Staff i gives the ranked preference list, P_i ={1,2,3}, as acceptable and {4,5,6} as the ranked unacceptable list, then to negotiate $(i,5)$, the i's preference list would need to be extended to {1,2,3,4,5}, making both Job 4 and 5 acceptable with strict preference of Job 4 to Job 5. By contrast, in *Append-to-End* the updated preference list would simply be {1,2,3,5}, leaving out Job 4 and making the stability conditions in *Append-to-End* easier to satisfy than in *Extend-Thru.* Below, Constraints (13) and (14) fulfill stability Constraint (9) in Formulation 2 for the *Extend-Thru* negotiation mechanism.

$$x_{ij} + \sum_{(k>_i j)\in A_i} x_{ik} + \sum_{(k>_i j)\in P_i\setminus A_i} y_{ik} + \sum_{(k>_j i)\in A_j} x_{kj} + \sum_{(k>_j i)\in P_j\setminus A_j} y_{kj} \geq 1,\ \forall (i,j)\in A \quad (13)$$

$$y_{ij} + \sum_{(k>_i j)\in A_i} x_{ik} + \sum_{(k>_i j)\in P_i\setminus A_i} y_{ik} + \sum_{(k>_j i)\in A_j} x_{kj} + \sum_{(k>_j i)\in P_j\setminus A_j} y_{kj} \geq 1, \forall (i,j)\in (I\times J)\setminus A$$
(14)

Under *Extend-Thru*, stability is upheld over the complete ranked preference lists. Constraint (13) ensures that no acceptable pairs block a solution, and Constraint (14) does the same for unacceptable pairs. Essentially, the difference between $minNegotiations$ under *Extend-Thru* and the regular stable matching problem with complete lists (Formulation 1) is that the objective is to minimize the number of unacceptable pairs chosen. Thus, this model can be thought of as a weighted stable matching problem with complete preference lists.

Due to the tighter stability constraints under *Extend-Thru* relative to those under *Append-to-End*, *Naive Algorithm* does not always produce feasible solutions to $minNegotiations$, since unacceptable pairs can block a solution. Thus, the problem can require more than $N - M$ negotiated pairings to achieve an almost acceptable complete stable matching. Fortunately, we show that the structure of *Extend-Thru* implies that a solution can be found in reasonable time using linear programming. Specifically:

Theorem 6. *The linear relaxation of minNegotiations (under the negotiation scheme Extend-Thru) has an integral polyhedron.*

Proof. The reader is referred to Proof Appendix A.3. □

Corollary 7. *minNegotiations (under the negotiation scheme Extend-Thru) can be solved optimally in polynomial time using the linear programming relaxation.*

Proof. Because the linear relaxation of *minNegotiations* (under the negotiation scheme *Extend-Thru*) has an integral polyhedron, optimal solutions to the relaxed problem are integral solutions to *minNegotiations.* Linear programming can be solved in polynomial time [15]. □

2.3.3 Minimizing the Cost of Negotiations

2.3.3.1 General Cost Function

To refine the matchings proposed by our models, since certain negotiations may be preferable to others in practice, we introduce the $minCostNegotiations$ problem. For almost acceptable complete stable matchings, Formulation 2 is given an updated objective function incorporating a cost function (15), to replace Objective Function (6) in the model. A wide variety of customizable objectives can be modeled for a given instance or context through this generic cost function.

$$min \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{(i,j)\in I\times J\setminus A} c_{ij}y_{ij} \tag{15}$$

For the negotiation scheme *Extend-Thru,* $minCostNegotiations$ has the same feasible region as $minNegotiations$ and can also be solved efficiently, as the following corollary summarizes.

Corollary 8. *minCostNegotiations (under the negotiation scheme Extend-Thru) can be solved optimally in polynomial time using the linear programming relaxation.*

For *Append-to-End*, since $minCostNegotiations$ cannot be solved in polynomial time through the linear programming relaxation (recall Proposition 2), an algorithmic approach is motivated for larger problem sizes. We next introduce a particular cost function and explore this topic further, developing a heuristic for $minCostNegotations$ under *Append-to-End.*

2.3.3.2 Minimizing Agent Negotiations

Count Agent Negotiations Cost Function Throughout this section, we focus on minimizing the number of negotiations with specific agents. Our motivation is that it may be less costly to negotiate for pairings when one staff or job already finds the other acceptable. For example, if $i \in P_j$ or if $j \in P_i$ for a pair $(i,j) \notin A$, then it only requires one agent-specific negotiation to take place with either i or j, rather than two if neither found the other acceptable. In the *Count Agent Negotiations Cost Function* (for Objective Function 15), c_{ij}, the cost on each pair (i,j), is defined as follows:

$$c_{ij} = \begin{cases} 0, & if\,(i,j) \in A \\ 2, & if\ i \notin P_j \text{ and } j \notin P_i \\ 1, & otherwise \end{cases} \tag{16}$$

Under the *Extend-Thru* Negotiation Scheme We have from previous results (Corollary 8) that under the negotiation scheme *Extend-Thru*, linear programming can be used to solve $minCostNegotiations$ with the *Count Agent Negotiations Cost Function* (objective 15 with costs 16) in polynomial time. This implies that in an application area where agents express rankings over the agents that they find unacceptable, we can efficiently find the negotiated complete stable matching in which the fewest individual agents need to be negotiated with.

Under the *Append-to-End* Negotiation Mechanism On the other hand, the linear programming relaxation of *minCostNegotiations* with the *Count Agent Negotiations Cost Function* under *Append-to-End* is not guaranteed to find an optimal solution (Proposition 2). *Append-to-End* is an appropriate negotiation scheme when rankings over unacceptable agents are not available or appropriate, as may be the case for very large instances where a full set of rankings is unrealistic. We introduce a heuristic for the problem, Algorithm *minAgentNegotiations*, characterize its worst-case performance, and perform computational testing to assess its solution time and quality.

Algorithm *minAgentNegotiations* In Algorithm *minAgentNegotiations* (detailed in Algorithm 2.2), the first step is to solve the problem instance to find which staff and jobs are matched in the stable matching without negotiations. The second step constructs a subgraph of the unmatched staff and jobs connected by their unacceptable edges with the following costs: 1 if one staff or job lists the other and 2 if neither lists the other on their respective preference lists. These costs correspond with the *Count Agent Negotiations Cost Function* (objective 15 with costs 16). Third, the minimum cost bipartite matching problem is solved on the subgraph to obtain the $N - M$ edges matching all of *unmatchedStaff* and *unmatchedJobs* with the fewest number of agent negotiations in the subgraph.

Worst-Case Performance -Algorithm *minAgentNegotiations* Before running computational experiments, it is useful to characterize the worst case performance of Algorithm *minAgentNegotiations*, which we can do analytically. In Theorem 9, we show that 100% is the worst case optimality gap for Algorithm *minAgentNegotiations*, a gap that is tight in some instances (e.g., as illustrated in Figure 4).

Theorem 9. *Algorithm minAgentNegotiations has a worst case optimality gap of 100%*

Algorithm 2.2 Heuristic for *Append-to-End* $minCostNegotiations$ under the Count Agent Negotiations Functions

1. First, obtain a stable matching to the problem instance. Many techniques are available (e.g. see [35, 38, 74]). Let M be the number of pairs matched.

2. Construct a subgraph, G, of unmatched staff and jobs as follows:

   (a) Construct an *unmatchedStaff* list of the $N-M$ staff members that are unmatched and an *unmatchedJobs* list of the $N-M$ jobs that are unmatched.

   (b) Let G be a bipartite graph with a node created for every element of *unmatchedStaff* on one side and a node for every element of *unmatchedJobs* on the other side.

   (c) For $(i,j) \in$ *unmatchedStaff* $\times$ *unmatchedJobs*, add edge (i,j) to G with cost, c_{ij} according to the *Count Agent Negotiations Cost Function.*

3. Obtain $N-M$ pairings in a matching of *unmatchedStaff* and *unmatchedJobs* by solving the minimum cost bipartite matching problem on G (where the minimum cost matching in G is found among those solutions with maximum cardinality, e.g., as in [6]) .

4. The final matching obtained by the algorithm is the union of the matchings obtained in Steps 1 and 4.

Proof. For a given instance of *minCostNegotiations* under the *Count Agent Negotiations Cost Function*, let z^* be the optimal solution value and let z^H be the solution value for *Algorithm minAgentNegotiations.* By Theorem 4, we have that $N-M$ total negotiations will be required for any instance. We have $N-M \leq z^* \leq 2(N-M)$ and $N-M \leq z^H \leq 2(N-M)$ since each negotiation arc in the optimal solution will cost at least 1 and at most 2. Therefore, $\frac{z^H - z^*}{z^*} \leq \frac{2(N-M)-(N-M)}{N-M} = 1$. We find a case where this bound is tight, illustrated in Figure

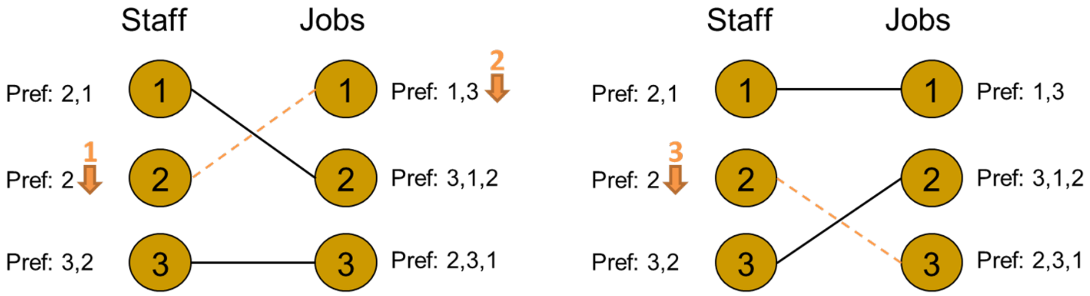


Figure 4: Example of the solution structure where Algorithm *minAgentNegotiations* has a worst case optimality gap of 100%.

4, where two agent-negotiations are required by Algorithm *minAgentNegotiations* (on the left) but only one is required in the optimal solution (on the right). □

Note, Step 2(c) in Algorithm *minAgentNegotiations* can be modified to assign a more general cost function over negotiations. For the case with random costs between 1 and 2, inclusive, on negotiation arcs, the worst-case bound of Theorem 9 will still apply.

Computational Experiments - Methods We next present the methodology used in our computational experiments to test Algorithm *minAgentNegotiations*. For each instance, we find the heuristic solution using Algorithm 2.2 and the optimal solution using IP (Formulation 2 with *Append-to-End*'s stability Constraint (12) and the *Count Agent Negotiations Cost Function*). We also compute the largest optimality gap possible per instance (Expression 17), where where z^H is the heuristic solution value and $N - M$ is used as a lower bound on the optimal solution cost per instance (exactly one agent-negotiation per $N - M$ total negotiations). All experiments were run using a 2.60 Ghz Xeon E5-2670 processor.

$$\frac{z^H - (N - M)}{N - M}(100\%) \tag{17}$$

We first run experiments showing the performance as the problem size (N, the number of staff members) increases. Problem sizes ranged from smaller assignments ($N \leq 100$) to the size of WFP reassignment exercise ($N = 500$) to the estimated size of an NMRP or monthly US Navy assignment instance ($N = 30,000$). For each problem size, 30 instances are run unless otherwise noted. In each instance, complete staff and job preference lists for each agent were randomly generated and then each was truncated to a random length according to a discrete, uniform distribution over [1,N].

Second, for $N = 100$, we fix a certain preference list length, k, for each agent in the network. For each k ∈{2, 5, 10, 20, ..., 90, 100}, we run 30 problem instances. In each, complete staff and job preference lists for each agent were randomly generated and then each was truncated to length k.

Third, we solve a generalization of Algorithm *minAgentNegotiations* where Step 2(c) is modified to a assign a more general cost function to negotiation arcs. In these experiments,

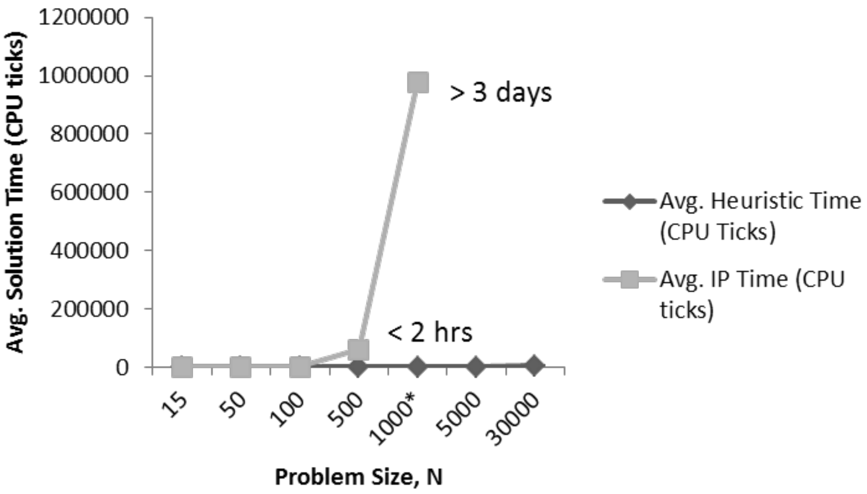


Figure 5: Average solution time comparison: Algorithm *minAgentNegotiations* vs. Integer Programming (*only five trials were run for $N = 1000$)

the problem size varies from $N = 15$ to $N = 30,000$, and negotiation arc costs are randomly assigned between 1 and 2, inclusive. For each problem size, 30 instances are run. In each instance, complete staff and job preference lists for each agent were randomly generated and then each was truncated to a random length according to a discrete, uniform distribution over [1,N].

Computational Experiments - Results In the first set of experiments, solution time and quality are studied as the problem size increases. Figure 5 plots the average solution time of Algorithm *minAgentNegotiations* and the Formulation 2 IP under *Append-to-End.* We see that the solution time of the optimal IP greatly increases from a problem size of $N = 500$ to $N = 1000$, while the solution time of Algorithm *minAgentNegotiations* remains relatively low for all problem sizes investigated. For the IP approach, solutions for $N = 500$ took just under 60,000 CPU ticks on average (approximately 2 hours) and for $N = 1000$, over 950,000 CPU ticks on average (over three days on average, and in some cases over 6 days). Due to the lengthy solution time, only five trials were run for $N = 1000$ for the IP. In contrast, even for for $N = 30,000$, Algorithm *minAgentNegotiations* took only 4,620 CPU ticks on average (approximately 70 minutes), making the algorithm tractable for even the largest stable matching instances in practice.

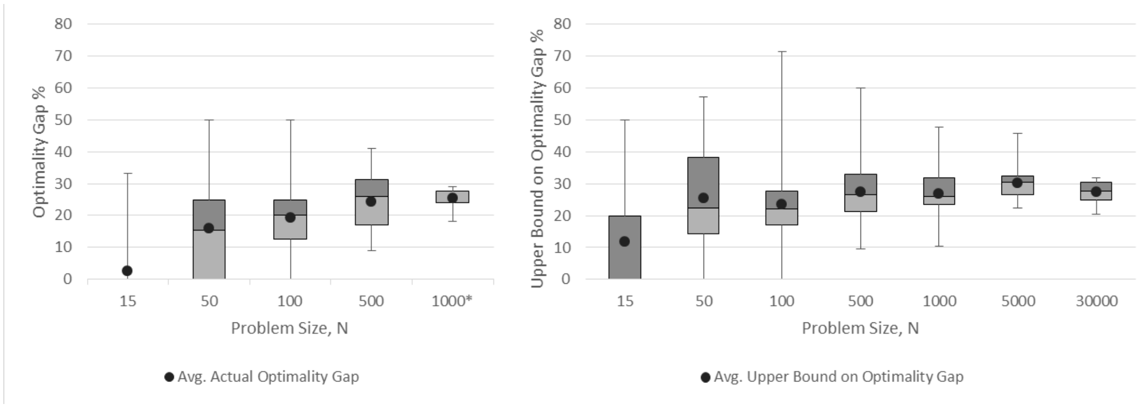


Figure 6: Box-and-whisker plots of the optimality gap of Algorithm *minAgentNegotiations* – Actual on the left and Upper Bound (according to Expression 17) on the right (*only five trials were run for $N = 1000$)

On the left in Figure 6 we have a box-and-whiskers plot of the optimality gap between Algorithm *minAgentNegotiations* and the Formulation 2 IP under *Append-to-End* for each problem size with the average value superimposed on top. The min and max are indicated by the whiskers, and the box indicates the median and first and third quartiles. Optimal solutions for comparison to heuristic solutions were generated up to $N = 500$ for all 30 random instances per size (and for 5 instances for $N = 1000$) due to the solution time required to solve larger IP instances.

The heuristic achieves the optimal in one or more cases, for $N \in \{15,\ 50,\ 100\}$. The worst-case gap is lower for the large problem sizes, although this could be due to limited number of instances run. The average optimality gaps were 2.4%, 16.0%, 19.3%, 24.3%, and 25.3%, corresponding to N =15, 50, 100, 500, and 1000, respectively. The average gap seems to increase with the problem size but may level off as the problem size becomes large.

Since optimal results were not generated for larger problems, we bound the largest optimality gap possible per instance on the right in Figure 6, according to Expression 17. The average optimality gap upper bounds were 11.7%, 25.5%, 23.6%, 27.5%, 26.8%, 30.2%, and 27.5%, corresponding to N =15, 50, 100, 500, 1000, 5000, and 30,000, respectively. We see that the average optimality gap upper bound seems to increase with the problem size but may level off as the problem size becomes large, and the worst-case gap is lower for the large problem sizes.

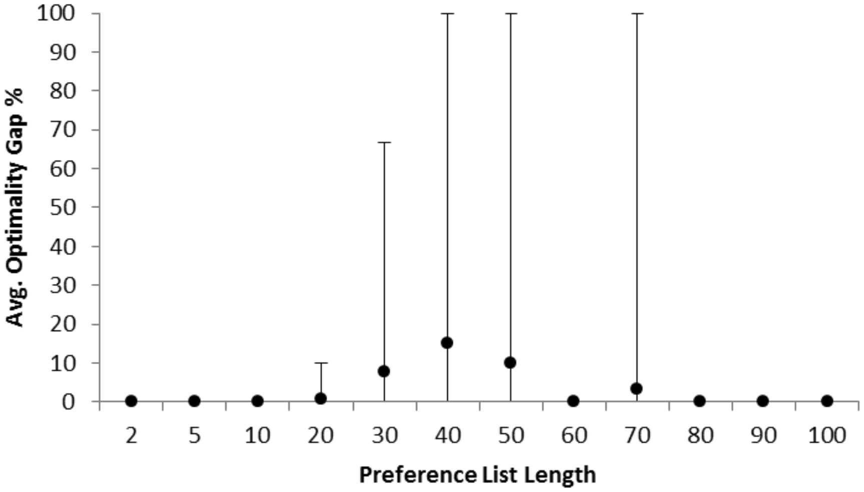


Figure 7: Optimality Gap Percentage as Preference List Length Increases for$N = 100$

In the second set of experiments, the length of agent preference lists varies. In Figure , we have a box-and-whiskers plot of the optimality gap between Algorithm *minAgentNegotiations* and the Formulation 2 IP under *Append-to-End* for each truncation length. In all cases, up to the third quartile in the data has an optimality gap of 0%, with the average varying from 0-15%, and the maximum optimality gap as high as the worst-case 100%. Algorithm *minAgentNegotiations* achieves the optimal solution in the instances studied for $k \leq 10$ and $k \geq 80$, with worse performance (6-15% average optimality gap) when preference lists range from between $k = 30$ and $k = 50$.

For the same instances as the preference list length varies, Figure 8 presents a box-and-whiskers plot of the optimal solution value to the Formulation 2 IP under *Append-to-End* for each truncation length. Here, we see that the minimum number of agent negotiations required for an instance decreases as agents increase the length of their preference lists. When $k = 2$, 95.77 agent negotiations are required. By $k = 20$, only 14.07 agent negotiations are required, and for $k \geq 30$, less than 5 agent negotiations are required.

In the third set of experiments, random costs between 1 and 2 are assigned for negotiation arcs. On the left in Figure 9 is a box-and-whiskers plot of the optimality gap between the modified Algorithm *minAgentNegotiations* and the Formulation 2 IP under *Append-to-End* for each problem size with the average value superimposed on top. The min and max are

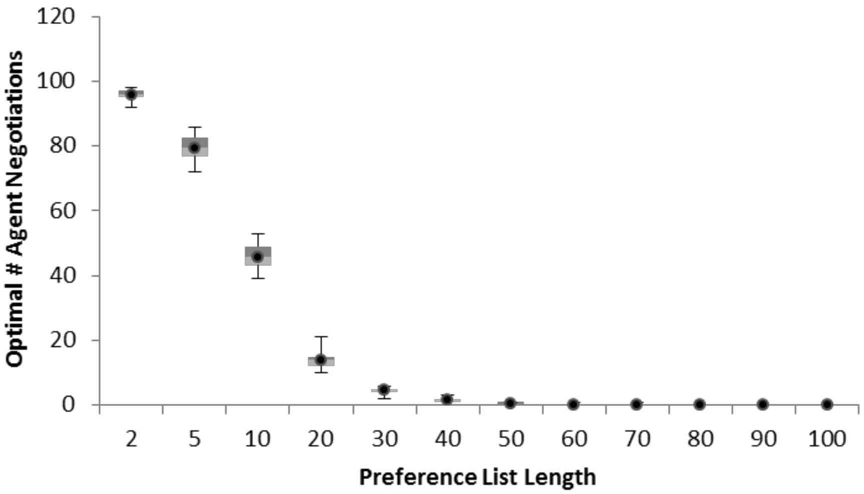


Figure 8: Optimal Solution Value as Preference List Length Increases for$N = 100$

indicated by the whiskers, and the box indicates the median and first and third quartiles. Optimal solutions for comparison to heuristic solutions were generated up to $N = 500$. The heuristic achieves the optimal in one or more cases, for $N \in \{15,\ 50,\ 100,\ 500\}$. The worst-case gap is lower for the large problem sizes, although this could be due to limited number of instances run. The average optimality gaps were 5.3%, 4.8%, 2.9%, and 1.5%, corresponding to N =15, 50, 100, and 500, respectively. The average gap seems to decrease with the problem size.

Since optimal results were not generated for larger problems due to solution times, we bound the largest optimality gap possible per instance on the right in Figure 9, according to Expression 17. The average optimality gap upper bounds were 32.6%, 19.5%, 14.5%, 6.3%, 5.1%, 2.3%, and 0.9%, corresponding to N =15, 50, 100, 500, 1000, 5000, and 30,000, respectively. We see that the average optimality gap upper bound decreases as the problem size becomes large, and the worst-case gap is also lower for the large problem sizes.

Computational Experiments - Further Discussion We additionally explore the performance difference between the heuristic and the optimal with an example. The IP can achieve a better solution than Algorithm *minAgentNegotiations* if the problem instance has an optimal solution to *minCostNegotiations* in which the acceptable arcs appearing are

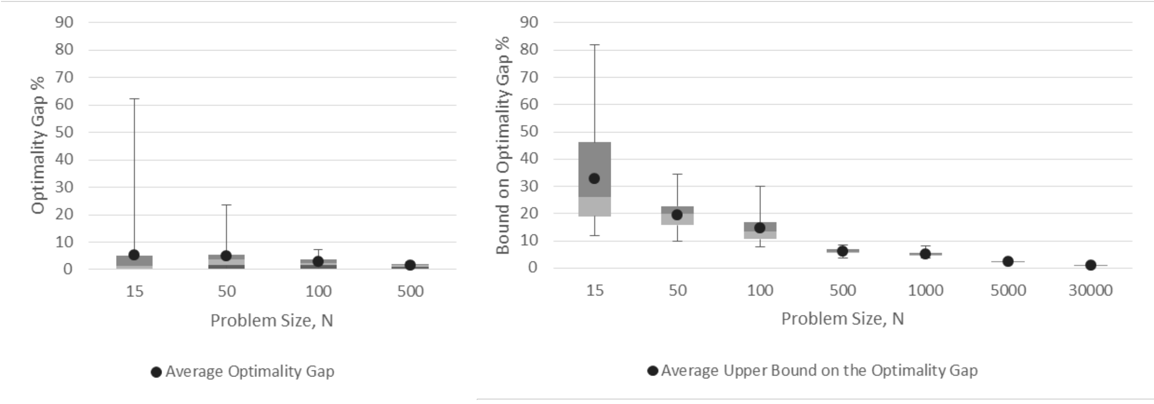


Figure 9: Box-and-whisker plots of the optimality gap of modified Algorithm *minAgentNegotiations* for general costs – Actual on the left and Upper Bound (according to Expression 17) on the right.

unstable according to the original preferences. While almost acceptable stable matchings are stable according to the negotiated preferences, not all are stable according to the original preferences. For example, in Figure 4, according to the original preferences, the acceptable arc matching from the solution on the right, {(1,1), (3,2)}, is unstable because (3,3) blocks it (Staff 3 prefers Job 3 to Job 2 and Job 3 prefers being matched to being unmatched). Algorithm *minAgentNegotiations* is unable to generate this solution because a subgraph consisting of Staff 2 and Job 3 can never be formed for this instance since all solutions to Algorithm *minAgentNegotiations* build upon stable matchings (according to the original preferences).

For small sized problems, solving the IP provides the best solution and can be done in a reasonable time. For large-sized instances (e.g., 1000 or more), we recommend using a heuristic. The one that we have described, Algorithm *minAgentNegotiations*, is simple to implement, and we find that the average optimality gap is 17.4% over the problems we studied in the first set of experiments and 3.05% in the second set of experiments for the *Count Agent Negotiation Function.* For the third set of experiments, with a more general cost function and generalization of the heuristic, the optimality gap was 3.63%.

For the problem sizes relevant in the humanitarian staff assignment context (e.g., approximately $N = 500$ for WFP), the IP will be solvable, even in software like Excel with an open-source optimization add-in. While the heuristic's performance is not as good, it is easy to implement even for very large problems. Further, in our experiments varying a fixed preference list length, the performance of Algorithm *minAgentNegotiations* was near-optimal for shorter and longer preferences lists and the number of agent negotiations required reduced as the length increased, suggesting that market design mechanisms (such as requiring a certain length of preference list from each agent) may be an additional approach to reducing an organization's negotiation costs.

2.3.4 Extension of the Negotiation Paradigm: Altering the Sequencing of Preferences

Our negotiation mechanisms can be enhanced by not only negotiating over unacceptable pairs, but also by negotiating for changes to the sequencing within staff and job rankings. This can in turn enable previously unstable, acceptable pairings to appear in negotiated complete stable matchings. The additional flexibility introduced by altering the sequencing of preferences allows for more possible matchings through negotiation. The two key properties of negotiated complete stable matchings are maintained under the extension: (i) that all staff and jobs are matched and (ii) that no blocking pairs exist according to the negotiated preference lists.

In this section, we extend the previous ideas by relaxing the assumption that only unacceptable pairings can be negotiated. Under the enhanced mechanism with reordering of preferences possible, we introduce negotiation scheme *Move-to-Beginning* which assumes that the incentives are strong enough so that whenever a pair (i, j) is selected for negotiation, i is moved to the top of j's preference list and j is moved to the top of i's preference list. This ensures that i and j are always paired together and cannot be involved in any blocking pairs. This negotiation mechanism can mimic the situation when a central matchmaker wishes to fix certain pairings within a matching and is willing to offer strong incentives to convince the involved staff member and job to modify their first preferences.

Move-to-Beginning modifies staff members' and jobs' top preferences by either moving an existing ranked staff or job to first preference or by appending a previously unacceptable staff or job to the beginning of a preference list. To negotiate for pair (i, j), a central matchmaker would need to offer strong incentives in order for i to become the first preference of j and for j to become the first preference of i. When $i \in P_j$ and $j \in P_i$ (prior to negotiations), the negotiation occurs over an acceptable pair, a unique feature of this negotiation scheme compared to the others investigated, which gives added flexibility. This modeling feature is based on staff and jobs, in actual negotiations, being open to reordering the ranking within their preference lists through negotiation in addition to wanting to receive incentives for being matched with an unranked staff or job.

The formulation *Move-to-Beginning* requires the introduction of z variables for negotiations on acceptable pairings. Defined only for acceptable pairs, $(i, j) \in A$, z_{ij} is a binary variable that is 1 if staff member i is assigned to job j (through a negotiation on (i, j)) and 0 otherwise. Formulation 3 gives the updated $minNegotiations$ model for *Move-to-Beginning*. The objective function (18) minimizes the number of total negotiations over unacceptable and acceptable pairs, and as in Formulation 2, a complete matching is required (Constraints (19) and (20)) according to a stability condition that takes into account the negotiation scheme (Constraint (21)) .

For the purpose of stability, staff and jobs are assumed to have indifferent preferences to unacceptable pairs, as in the case of *Append-to-End*. Therefore, stability in this model requires only that acceptable pairs do not block the solution matching, which Constraint (21) ensures. In the constraint, for each acceptable (i, j) either (a) i is paired with a job preferred at least as much as j or with a negotiated job that is now a top preference or (b) j is paired with a staff member preferred at least as much as i or with a negotiated staff member that is now a top preference. In either case, the pair is preventing from being a blocking pair. In the constraint, the summations involving the x variables are over preferred staff and jobs that are part of acceptable arcs, while the y and z variables are over the entire respective sets for which they are defined (since any negotiation involving staff i or job j in *Move-to-Beginning* automatically ensures the agent is getting the top choice and cannot be involved in a blocking pair).

We show that $minNegotiations$ for the negotiation scheme *Move-to-Beginning* always produces a feasible, negotiated complete stable matching, since its polyhedron is non-empty (see Corollary 10). However, we also show that for *Append-to-End*, the linear relaxation of

Formulation 3 *minNegotiations* Formulation for *Move-to-Beginning*

$$
\begin{array}{llll}
min & \sum_{(i,j)\in I\times J\setminus A} y_{ij} + \sum_{(i,j)\in A} z_{ij} & & (18) \\
s.t. & \sum_{j\in A_i}(x_{ij}+z_{ij}) + \sum_{j\in J\setminus A_i} y_{ij} = 1 & \forall i \in I & (19) \\
 & \sum_{i\in A_j}(x_{ij}+z_{ij}) + \sum_{i\in I\setminus A_j} y_{ij} = 1 & \forall j \in J & (20) \\
 & x_{ij} + \sum_{(k>_i j)\in A_i} x_{ik} + \sum_{k\in J\setminus A_i} y_{ik} + \sum_{k\in A_i} z_{ik} & & \\
 & + \sum_{(k>_j i)\in A_j} x_{kj} + \sum_{k\in I\setminus A_j} y_{kj} + \sum_{k\in A_j} z_{kj} \geq 1 & \forall (i,j) \in A & (21) \\
 & x_{ij},\ z_{ij} \in \{0,1\} & \forall (i,j)\in A & (22) \\
 & y_{ij} \in \{0,1\} & \forall (i,j) \in (I\times J)\setminus A & (23)
\end{array}
$$

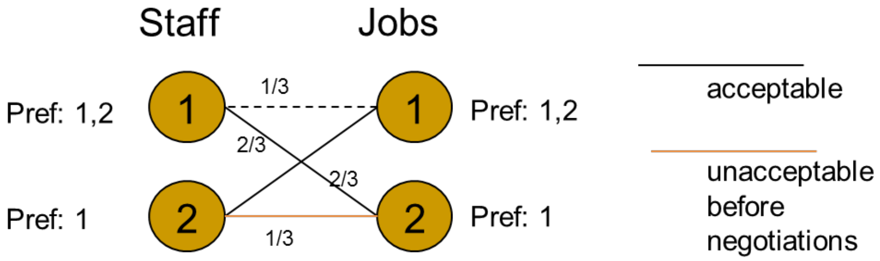


Figure 10: Fractional optimal solution to *minNegotiations* under *Move-to-Beginning*

minNegotiations does not have an integral polyhedron (Proposition 11).

Corollary 10. *There is a non-empty feasible region for minNegotiations under Move-to-Beginning.*

Proof. For a given instance of staff and job preferences, let $z = 0$ and let (x, y) be a feasible, integer solution to *minNegotiations* under *Append-to-End,* which we know to exist by Theorem 1. Because (x, y) satisfies the constraints of Formulation 2 (Constraints (7)-(11) and (12), (x, y, z) clearly satisfies the constraints of Formulation 3 (Constraints (19)-(23)). Thus, we have a feasible, integer solution to *minNegotiations* under *Move-to-Beginning.* □

Proposition 11. *The linear relaxation of minNegotiations (under the negotiation scheme Move-to-Beginning) does not have an integral polyhedron.*

Proof. This proof is by counterexample. Figure 3 shows a fractional, optimal vertex solution to *minNegotiations* under *Move-to-Beginning* that was found using the simplex algorithm implementation in Gurobi Optimizer 5.6 *[61].* □

As in *Append-to-End*, we show that again surprisingly $N - M$ negotiated pairs are required under *Move-to-Beginning* to create a negotiated complete stable matching (Theorem 12). This is true despite *Move-to-Beginning* being a stronger negotiation tactic than *Append-to-End* in that more blocking pairs are eliminated per negotiation, which in turn causes Constraint (21) to be less restrictive than Constraint (12). The proof hinges on bounding the increase in the cardinality of pairs matched as a result of any single negotiated pairing. The technique achieves the bound by examining the impact of a negotiated pairing (and the corresponding updated preferences) under a structured order of staff member proposals in the classic Gale-Shapely stable matching algorithm [35] and using the fact that all possible executions of the algorithm yield the same stable matching same solution [38].

Theorem 12. *The optimal solution value of the minNegotiations is N-M under the Move-to-Beginning negotiation scheme.*

Proof. Since *Naive Algorithm* results a feasible solution to $minNegotiations$ under *Move-to-Beginning* for any problem instance, $N - M$ is an upper bound for the model. We also have that $N - M$ is also a lower bound for the solution value to $minNegotiations$ under *Append-to-End* (see Proof Appendix A.4 and specifically Lemma 34). Thus, we can conclude that $minNegotiations$ under *Append-to-End*'s optimal solution value is $N - M$. □

The negotiation mechanism in *Naive Algorithm* can be revised for *Move-to-Beginning* by simply by modifying that (i, j) is negotiated into the matching by moving/appending i to the front of P_j and j to the front of P_i (in Step 3b in Algorithm 2.1), and it is easily seen to still require $N-M$ negotiated pairings. Thus, as in the *Append-to-End* negotiation scheme, *Naive Algorithm* is shown to produce an optimal solution to $minNegotiations$ (Corollary 13), and thus *Naive Algorithm* proves useful in achieving polynomial-time, minimal negotiation solutions.

Corollary 13. *Naive Algorithm achieves a minimum negotiation solution to minNegotiations under the Move-to-Beginning negotiation scheme.*

Proof. From Theorem 12 we have that the optimal solution value of the *minNegotiations* is

$N-M$ for *Move-to-Beginning. Naive Algorithm* produces an $N-M$ negotiation feasible solution to *minNegotiations.* □

Despite all solutions involving $N-M$ negotiated pairings for *minNegotiations* under the negotiation scheme *Move-to-Beginning*, it can be observed that different solutions can have different numbers of unacceptable pairings appearing in the final matching. For example, for the matching instance from Figure 1, the *Naive Algorithm* output for the instance, {(1,1), (2,3), (3,2)}, contains one negotiated unacceptable pair. On the other hand, an alternative solution for the instance with no unacceptable pairs can be found, {(1,2), (2,3), (3,1)}, by negotiating for pair (3,1) by reordering Job 1's preference list from $P_1 = \{1,3,2\}$ to $P_1 = \{3,1,2\}$. For negotiated complete stable matchings, Formulation 3 can also be given an updated objective function incorporating a cost function (24), leading to interesting future research directions in *minCostNegotations* for *Move-to-Beginning* and other negotiations schemes that may be possible through altering the sequencing of staff or job preferences.

$$min \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{(i,j)\in I\times J\setminus A} c_{ij}y_{ij} + \sum_{(i,j)\in A} \bar{c}_{ij}z_{ij} \tag{24}$$

2.4 Conclusion

Our research contributes to the growing stream of studies analyzing decentralized markets. At a high level the research in this stream discusses equilibrium concepts and the design of mechanisms for achieving socially desirable outcomes. The importance of the subset of this research on two-sided-markets was recently emphasized by the awarding of the Nobel Prize for the algorithms that ensure stable matchings in a broad set of contexts from school choice to kidney allocation. Yet much remains for solving practical large-scale problems that require complete stable matchings. The objective of this research was to develop mathematical models that can ensure a complete matching even when preference lists are truncated, where the matchings are stable and the models are scalable for large organizations.

We modeled negotiations, which occur in practice, as part of the problem of matching all agents. We provided mathematical programming formulations that result in negotiated complete stable matchings, minimizing the number or cost of negotiations, in which (i) all staff and jobs are matched and (ii) no blocking pairs exist according to the negotiated preference lists. Two negotiation schemes*,* *Append-to-End* and *Extend-Thru*, were specifically investigated.

When the centralized objective is to minimize the number of negotiations required, we showed that under *Append-to-End*, $N - M$ negotiations are required. Here, N is the problem size and M is the number of pairs matched in a stable matching for the instance prior to negotiations, and we also developed a polynomial-time *Naive Algorithm* that achieves an optimal solution. Under *Extend-Thru*, we showed that the problem can be solved optimally through linear programming. Compared to the existing literature on 'almost stable' matches, we found that our 'almost acceptable' mechanisms may require significantly fewer compromises to reach a complete matching (e.g., only 37 negotiated pairings instead of 400 blocking pairs).

When the centralized objective is to minimize the cost of negotiations, we introduced a generic cost minimization objective function and provided specific analysis for the *Count Agent Negotiations Cost Function,* which minimizes the number of negotiations with specific agents. Under *Extend-Thru*, and as generalizes for any linear cost objective for this negotiation scheme, we showed that the problem can be solved in polynomial time. Under *Append-to-End*, we found that a linear relaxation of the IP is not guaranteed to produce an optimal solution, and we introduced a heuristic with fast solution times, even for very large problems, that is simple to implement.

Last, we extended our concept of negotiation modeling to include negotiating for changes to the sequencing within staff and job rankings. We introduced the negotiation scheme, *Move-to-Beginning,* which assumes that negotiation incentives are strong enough so that whenever a pair is selected for negotiation, the staff member and job in the pair both move each other to the front of their respective preference lists. We introduced an IP formulation for minimizing the number of negotiations required under *Move-to-Beginning.* As in *Append-to-End*, we showed that again surprisingly $N-M$ negotiated pairs are required under *Move-to-Beginning* to create a negotiated complete stable matching (which can be found through a variant of the *Naive Algorithm*). This is true despite *Move-to-Beginning* being a stronger negotiation tactic than *Append-to-End* in that more blocking pairs are eliminated per negotiation.

Our findings regarding negotiated and/or locally-stable matchings contribute to scientific knowledge at the intersection of optimization, computer science, and economics while at the same time making large-scale complete stable matchings more feasible. We build new models, provide structural results about the number of negotiations needed to achieve negotiated complete stable matches, create scalable approaches to solve problems efficiently, and analyze the performance of the algorithms. Overall, our results could influence the way that many decisions are made for industries that regularly reassign staff and jobs. We focus in this chapter on matching staff and jobs, but if the work is expanded to other application areas, it could lead to further theoretical results.

2.4.1 Future Directions

Several research directions are promising building on this work. One area of interest is to examine more general approaches to *minCostNegotiations.* Additionally, introducing market design mechanisms (such as requiring all agents to submit preference lists of a certain length) may be another approach to impacting the number or cost of negotiations required.

Moreover, within the scope of altering the sequencing within preference lists through negotiation, two potential additional negotiation schemes are (i) to swap agent positions on a preference list or (ii) to move an agent to a spot besides the front of the list. These schemes offer more flexibility than *Move-to-Beginning*, but with this added flexibility, modeling challenges may arise. This might introduce the opportunity to investigate other algorithmic approaches to achieving negotiated complete stable matchings.

Our negotiation framework can also be expanded further. A promising direction appears to be locally-stable matchings based on underlying social interaction networks. Here, analysis can be conducted on the trade-off between relaxing stability and negotiating over preferences. Finally, as motivated by the multiple reassignments over the course of a career of staff members at WFP or in the military context, our negotiation approach can be applied to dynamic assignments where stability is defined over multiple periods and uncertainty over future periods exists.

Chapter III

MANAGING BOTTLENECKS IN PORT AND OVERLAND TRANSPORT NETWORKS FOR HUMANITARIAN AID

3.1 Introduction

An uncertain environment with disruptions is a reality faced by many humanitarian operations. Delays in the transportation process from the port through the corridor prevent aid from reaching beneficiaries when needed. In addition, delays can also be quite costly, as vessel delays can be charged large demurrage fees and inland delays often require storage and handling fees. These delay costs consume resources that could potentially be used for increased procurements of aid. At the port, delays can be caused by too many vessels arriving too closely together or by not having enough bagging machines (which are often used to bag bulk or break bulk cargo before it is loaded onto trucks for offtake) or berths to meet the need. In the corridor, delays can occur at the beginning transition from the port bagging machines to trucking due to port storage/loading silos being full or to not enough trucks being available when needed, a problem that is sometimes attributed to strikes, government regulations, general transport shortages, and port entry delays (especially of foreign-owned vehicles). Corridor delays can also occur during the transportation leg, either due to physical damage to roads (e.g. rainy season or hazardous conditions) or customs/border crossings.

Clearly, catastrophic disruptions in the supply chain (e.g. major strikes or natural disasters) can be concentrated at the port and need to be mitigated as much as possible. However, smaller-scale and more frequent disruptions like customs and transport availability have similar implications on the supply chain. In a book published by the World Bank addressing logistics costs and supply chain reliability for landlocked countries, transport bottlenecks in the port and corridor are highlighted [9], and in particular they find that for normal operations (i.e. not during a major strike or disaster) the “most important source of delay is initiating transit in ports.” In this chapter, we address a practical set of related problems:

(i) Without precise data and in an environment of uncertainty, how can delays and congestion be modeled?

(ii) Given a characterization of congestion and delays, how can routing choices minimized the cost of delay?

(iii) With only a limited budget for improvements, where should investments be made in the transport network to reduce delays?

We first introduce a queuing-based model for quantifying congestion at ports and corridors. We obtain closed-form expressions for expected waiting time in the system, even though stochastic server breakdowns are incorporated. We next show that our delay function is convex with respect to flow, introduce a convex cost flow model that can be used to minimize delay, and give optimality conditions for minimizing port and corridor congestion delays for a structured network type. Finally, we characterize the monotonic impact of parametric changes on total wait, and we formulate a mathematical program that simultaneously invests a budget and routes flow optimally through the network.

Throughout, we compare solutions using the optimal approach to rules of thumb and identify important factors that might be missing in practical decision making currently. Overall, we create models that do not require precise or extensive inputs and for which most of the realistic-sized instances evaluated could be solved quickly and with open-source optimization (with the demo solvers included with GAMS and/or through the NEOS Server [54]).

This chapter is organized as follows. Section 3.2 highlights the contributions of this research and summarizes related literature. In Section 3.3, we introduce a delay model and relevant closed-form expected waiting time expressions. In Section 3.4, we incorporate the delay functions into a convex cost routing model evaluate the policy of routing in proportion to the effective processing rate on a path. Section 3.5 presents a resource allocation investment model to to improve network parameters and evaluates two contrasting rule-of-thumb investment policies. We conclude in Section 3.6 with a summary of the work and directions for future research.

3.2 Literature Review and Contributions

Our research makes contributions to several areas in the literature. We develop new humanitarian transport models that are not based on assumed knowledge of deterministic inputs. We incorporate smaller-scale sources of uncertainty in the corridor instead of focusing on larger-scale disruptions at the port. We incorporate congestion and breakdowns into a routing model, which is thus far unaddressed in the humanitarian context, and introduce a pseudopolynomially solvable convex cost routing model. Finally, we develop a model that simultaneously consider capacity expansion, potential improvements to failure parameters, and routing in a network to minimize expected waiting time under uncertainty, a scope not shared by any existing papers in the literature to the best of our knowledge.

In humanitarian transport models, most existing models require precise and systematic data, like their counterparts in more traditional private sector operations (such as [4, 8, 14, 31, 42, 81]). For example, Berkoune et al [14] examine detailed vehicle routing in disaster response operations, and Alvarenga et al [8] examine East African corridor optimization for WFP through port simulation in conjunction with a time-expanded, multicommodity flow model. However, in practice, there is often a lack of detailed real-time data, and this is further exacerbated by the uncertain environment in which humanitarian operations occur where security [12], natural disasters [11], or strikes and customs issues [9] undermine the assumption of deterministic costs and capacity over time. Here, our models are a contribution by requiring fewer inputs and allowing for estimated values that can capture essential features of the port and corridor transportation network and predict the impact of congestion delays.

Ports and the transition to land transport play crucial role in international trade and logistics, and bottlenecks and disruptions negatively impact the flow of humanitarian aid [82]. Catastrophic disruptions at the port have historical precedence and wide-reaching impact (e.g. the notable ten-day shutdown of 29 west-coast ports in the United States in 2002 which caused months of cargo backlog [64]). Mitigation and inventory planning are two areas that have been studied for dealing with major port disruptions. In [46], the authors model port disruptions with a discrete-time Markov chain and focus on mitigation strategies, such as contingency rerouting plans, emphasizing the need for capacity expansion and contingency rerouting. Inventory management for risk mitigation in the face of port-of-entry closures is addressed in [45] through an infinite-horizon, periodic-review inventory control model. However, smaller-scale and more frequent disruptions like customs and transport availability have similar implications on the supply chain [9], and we have not seen these addressed in humanitarian transport literature. Our modeling paradigm addresses these smaller-scale sources of uncertainty in the corridor as a stochastic disruption process that can be defined for different levels of frequency, duration, and variability of repair by adjusting parameters.

We further model how to manage flow when multiple port and transport network options are available. Limited models exist for incorporating congestion delays into humanitarian transport models. Zhang et al [92] address bottlenecks with respect to location modeling, but routing decisions are not addressed. More generally, congestion in transportation networks has been addressed in [18, 22, 29], but these models assume an underlying time-expanded network flow model, requiring many deterministic inputs and not adequately incorporating breakdowns when all flow is halted due to a disruption. Our work contributes by modeling congestion in a transport network with breakdowns and without requiring the deterministic inputs needed for a time-expanded network flow approach.

Our queuing approach to delay modeling leads to nonlinear expected waiting time functions, which we show to be convex with respect to routing decisions. Nonlinear optimization and queuing models is addressed in [23], but their work does not include breakdowns (or general service distributions) and their network analysis is limited to a closed network of queues in which a fixed number of customers or tasks circulate indefinitely. Our work contributes a pseudopolynomially solvable convex cost routing model for open queuing networks of a certain assumed structure that incorporate disruptions.

Lastly, we consider a network design problem of where and how much to invest a limited budget in network improvements. Several streams of work are related. In [62], the authors consider the problem of expanding arc capacities in a network in a robust formulation considering demand and travel time uncertainty, while Ahmed et al [5] present a multi-stage stochastic integer programming approach for capacity expansion under uncertainty. In [7], the authors investigate simultaneous optimization of capacity and planned lead time in production system, and Stidham [83] gives an overview of design and control of queuing systems. We develop a model that simultaneously considers capacity expansion, potential improvements to failure parameters, and routing in a network to minimize expected waiting time under uncertainty – a scope not shared by any existing papers in the literature to the best of our knowledge.

3.3 Modeling Port and Corridor Delays

We first introduce a queuing-based model for predicting congestion in ports and corridors and obtain closed-form expressions for expected waiting time in the system. Then, we show that these waiting time expressions are convex with respect to the arrival rate. Finally, we provide computational-based insights into congestion models and highlight the importance of being mindful of variance of corridor downtime in the network.

3.3.1 Port and Corridor Queuing Delay Model

At discharge ports, aid can be delayed when vessels queue at sea due the arriving vessels exceed the port capacity (berth space, bagging machines, etc.). Aid can also be delayed once cargo is on land as unloaded tonnage awaits offtake into the corridor or inland due to customs delays or road outages [86, 82, 50]. Offtake delays into the corridor can occur due to capacity restrictions (e.g., the trucking tonnage contracted by a humanitarian organization for the month) and also due to disruptions (e.g., security issues, limited truck access to the port, and even labor strikes) [67, 9, 24]. We model delays through a two station, tandem queuing model (as pictured in Figure 11) where the first station models delays at the port and the second station models delays in the corridor, with the inclusion of stochastic breakdowns.

In the humanitarian context, regardless of the inland delivery point, offtake delays and disruptions often occur collectively at the port (e.g., while waiting for access to transport or paperwork to clear [86, 82, 50]). We capture these inherent breakdowns and delays at

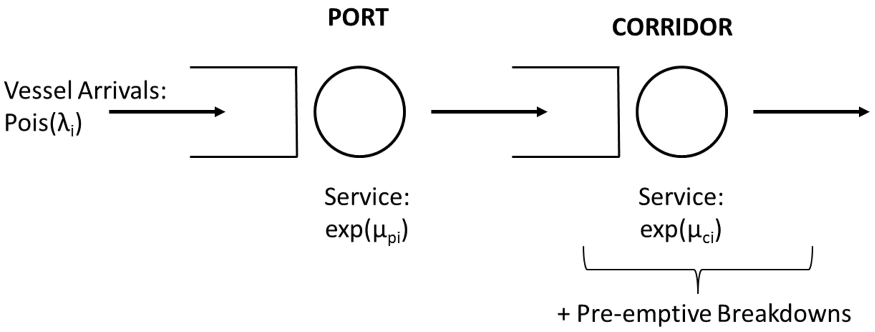


Figure 11: Two station, tandem queuing model for port and corridor delays

the port, which are an important source of delay in humanitarian aid, by assuming that all flow from the port proceeds into the corridor delay station. For this structured case, notation is specified in terms of port-corridor pair i, where the port station i feeds into the corridor station i. Table 2 summarizes the notation used in the delay model. Our results also generalize for other flow assumptions and network configurations, as will be discussed in later sections for routing and investment models, as long as the arrival and service distribution assumptions hold at each station.

Table 2: Notation Reference for the Port-Corridor Delay Model

Notation	**Meaning**
$\lambda_i \geq 0$	Vessel arrival rate to port-corridor pair i, which follow a Poisson process
$\mu_{pi} > 0$	Exponential processing rate at port i (capturing berths, bagging machines, and other port factors)
$\mu_{ci} > 0$	Exponential processing rate at corridor i
$f_i > 0$	Mean time to failure in corridor i, according to a Poisson failure process with rate $\frac{1}{f_i}$
$r_i \geq 0$	Mean time to recovery in corridor i after a failure
$v_i \geq 0$	Variance for recovery time in corridor i
A_i	Long-run availability of corridor i ($\frac{f_i}{(f_i+r_i)}$)
ρ_{pi}	Utilization at port i
ρ_{ci}	Utilization at corridor i
W_{pi}	Expected time per vessel at port i in queue and in service
W_{ci}	Expected time per vessel at corridor i in queue and in service
W_i	Total expected time per vessel at port and corridor i ($W_{pi} + W_{ci}$)

Instead of focusing on what should be done in managing daily operations or on exceptional time periods where the port sees a rapid scale-up of activity or a lengthy closure, the chosen model captures a high-level, strategic analysis of where and how lengthy the delays in the network will be. The resulting long-term characterization of expected delays allows humanitarian logisticians to better understand and plan for systemic delays and incorporate them into the decision of routing of aid through the ports. Such long-term analysis of on-going operations is relevant for humanitarian organizations; for example, a majority of the food procurements at the UN World Food Programme that traveled overseas went toward non-emergency projects in 2011 [68].

The first station in the tandem queuing system models port delays as an M/M/1 queue. A single server at the port is assumed since the objective is not to capture the specific movements in and out of the berths and through the bagging machine stations, but rather the overall delay time spent at the port. If the station is stable (i.e., if $\rho_{pi} = \frac{\lambda_i}{\mu_{pi}} < 1$), then the expected time at the port in queue and in service, W_{pi}, is given by Equation 25 as shown in Theorem 14.

$$W_{pi} = \frac{1}{(\mu_{pi} - \lambda_i)} \tag{25}$$

Theorem 14. *Assuming station stability ($\lambda_i < \mu_{pi}$), the expected time at the port in queue and in service is $W_{pi} = \frac{1}{(\mu_{pi} - \lambda_i)}$.*

Proof. For an M/M/1 queue with Poisson process arrivals at rate λ_i and exponential service rate μ_{pi}, the expected waiting time in queue and in service is known to be $\frac{1}{(\mu_{pi} - \lambda_i)}$ (see Eq. 4.2.14 in [16]). □

Departures from the port i station are assumed to proceed directly into the queue at the corridor i station. For the corridor, the addition of stochastic failures to the offtake server causes the station to be an M/G/1 queue. Stochastic failures are assumed to be preemptive (meaning that a disruption can occur in the middle of service and service will be preempted until the server is working again).

We model these "corridor breakdowns" according to the preemptive failure modeling framework described in [40] that uses f_i, r_i, and v_i and is visualized in Figure 12. We assume failures are exponentially distributed, with f_i as the mean time to failure (the inverse of this

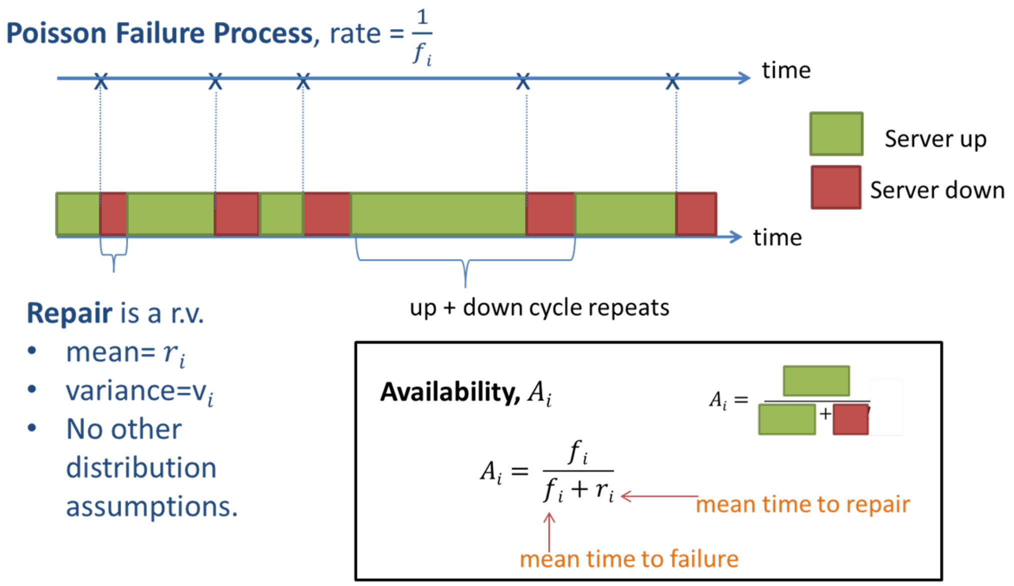


Figure 12: Preemptive breakdown model illustration. Three parameters, f_i, r_i, and v_i, characterize breakdowns at Corridor i.

is the rate for the failure process). When a failure occurs, the server is down and unavailable. We assume knowledge of the first and second moment of the time the server is down, with r_i as the mean time to repair and v_i as the variance of the repair time, but we do not assume a known distribution on the repair time. By using a single server with this breakdown model, we cover the case in which all of the corridor transport is down at once, which would occur in the case of a transport strike, security incident or service outage from the shipper's carrier base.

The long-run proportion of time that the corridor is available for transport, represented as A_i, is then $A_i = \frac{f_i}{(f_i+r_i)}$. A_i can be used to verify that the parameters used for a specific port-corridor network match the real-life performance. If the station is stable (i.e., if $\rho_{ci} = \frac{(f_i+r_i)\lambda_i}{f_i\mu_{ci}} = \frac{\lambda_i}{A_i\mu_{ci}} < 1$), then we can derive the expected time at the corridor in queue and in service, W_{ci}, as is given by Equation 26 as shown in Theorem 15.

$$W_{ci} = \frac{2(f_i + r_i) + (r_i^2 + v_i)\lambda_i}{(2f_i\mu_{ci} - 2(f_i + r_i)\lambda_i)} \tag{26}$$

Theorem 15. *Assuming station stability ($\lambda_i < A_i\mu_{ci}$), the expected time at the corridor station in queue and in service is $W_{ci} = \frac{2(f_i+r_i)+(r_i^2+v_i)\lambda_i}{(2f_i\mu_{ci}-2(f_i+r_i)\lambda_i)}$.*

Proof. The expression for the expected waiting time in queue and in service for an M/G/1 queue is known to be $E[S] + \frac{\lambda E[S^2]}{2(1-\rho)}$, where S is a random variable for the service time at the station (in this case including the processing time and any preemptions), λ is the arrival rate, and ρ is the utilization (see Eq. 5.2.42 in [16]).

We have $E[S] = \frac{1}{A_i\mu_{ci}} = \frac{f_i+r_i}{f_i\mu_{ci}}$, $\rho = \frac{\lambda_i}{A_i\mu_{ci}} = \frac{(f_i+r_i)\lambda_i}{f_i\mu_{ci}}$, and $var[S] = \left(\frac{1}{A_i\mu_{ci}}\right)^2 + \frac{(r_i^2+v_i)(1-A_i)(1/\mu_{ci})}{A_i r_i} = \frac{2(f_i+r_i)^2+f_i(r_i^2+v_i)\mu_{ci}}{f_i^2\mu_i^2}$ (from Section 8.4.2 on Variability from Preemptive Outages in [40]).

Then, $E[S^2] = var[S] + E[S]^2 = \frac{2(f_i+r_i)^2+f_i(r_i^2+v_i)\mu_{ci}}{f_i^2\mu_{ci}^2}$.

Thus we have $W_{ci} = E[S] + \frac{\lambda E[S^2]}{2(1-\rho)} = \frac{2(f_i+r_i)+(r_i^2+v_i)\lambda_i}{(2f_i\mu_{ci}-2(f_i+r_i)\lambda_i)}$. □

The total expected delay per arrival to the port-corridor i is $W_i = W_{pi} + W_{ci} = \frac{1}{(\mu_{pi}-\lambda_i)} + \frac{2(f_i+r_i)+(r_i^2+v_i)\lambda_i}{(2f_i\mu_{ci}-2(f_i+r_i)\lambda_i)}$. If estimates can be made for μ_{pi}, μ_{ci}, f_i, r_i, and v_i, this closed-form expression could be a used by humanitarian logisticians to add expected delay time on top of assumed lead times for an operation, which might otherwise only include best-case processing times without factoring in the impact of congestion.

Further, the station delay expressions can generalize beyond the context of the presented tandem queuing network where all flow from a port feeds into the corridor delay station, as long as the arrival and service distribution assumptions hold. For example, departures from two port stations (with independent arrivals) could merge into a single Poisson process arriving to an inland border of a landlocked country, where delays and breakdowns in service can occur in dealing with customs and border-control can be modeled with a corridor delay station. Our delay models assume independent Poisson process arrivals to all stations and that stations with breakdowns are the last station in any path or series. This generalization is used in later sections exploring routing and investment decisions.

3.3.2 Structure of the Wait Function with Respect to Arrival Rate

A key concept in queuing theory is that delays do not scale linearly with the arrival rate. Instead, each additional arriving vessel has an increasing marginal delay cost. More formally, the expected station wait per vessel (W_{pi} and W_{ci}) and the total expected station wait ($\lambda_i W_{pi}$ and $\lambda_i W_{ci}$) are increasing and convex with respect to λ_i, subject to stability, which we next show in Theorem 16). We then have the corollary that expected total wait per vessel per path ($W_i = W_{pi} + W_{ci}$), expected total wait per path ($\overline{W_i} = \lambda_i W_i$), and expected total wait in the network ($\overline{W} = \sum_i \overline{W_i}$) are increasing and convex with respect to λ_i.

Theorem 16. *W_{pi}, $\lambda_i W_{pi}$, W_{ci}, and $\lambda_i W_{ci}$ are all strictly increasing and strictly convex with respect to λ_i, subject to stability conditions ($\lambda_i < \mu_{pi}$ and $\lambda_i < \frac{f_i \mu_{ci}}{(f_i + r_i)}$).*

Proof. Part 1: W_{pi}

$\frac{\partial W_{pi}}{\partial \lambda_i} = \frac{1}{(\mu_{pi} - \lambda_i)^2} > 0$ when $\lambda_i < \mu_{pi}$. Thus, W_{pi} is strictly increasing w.r.t. λ_i.

$\frac{\partial^2 W_{pi}}{\partial \lambda_i^2} = \frac{2}{(\mu_{pi} - \lambda_i)^3} > 0$ when $\lambda_i < \mu_{pi}$. Thus, W_{pi} is strictly convex w.r.t. λ_i.

Part 2: $\lambda_i W_{pi}$

$\frac{\partial \lambda_i W_{pi}}{\partial \lambda_i} = \frac{\mu_{pi}}{(\mu_{pi} - \lambda_i)^2} > 0$ when $\lambda_i < \mu_{pi}$, since $\mu_{pi} > 0$. Thus, $\lambda_i W_{pi}$ is strictly increasing w.r.t. λ_i.

$\frac{\partial^2 \lambda_i W_{pi}}{\partial \lambda_i^2} = \frac{2\mu_{pi}}{(\mu_{pi} - \lambda_i)^3} > 0$ when $\lambda_i < \mu_{pi}$, since $\mu_{pi} > 0$. Thus, $\lambda_i W_{pi}$ is strictly convex w.r.t. λ_i.

Part 3: W_{ci}

$\frac{\partial W_{ci}}{\partial \lambda_i} = \frac{2f_i^2 + 2r_i^2 + f_i r_i (4 + \mu_{ci} r_i) + f_i \mu_{ci} v_i}{2(f_i \mu_{ci} - (f_i + r_i)\lambda_i)^2} > 0$ when $\lambda_i < \frac{f_i \mu_{ci}}{(f_i + r_i)}$, since $\mu_{ci} > 0$ and all parameters are non-negative. Thus, W_{ci} is strictly increasing w.r.t. λ_i.

$\frac{\partial^2 W_{ci}}{\partial \lambda_i^2} = \frac{(f_i + r_i)\left(2f_i^2 + 2r_i^2 + f_i r_i (4 + \mu_{ci} r_i) + f_i \mu_{ci} v_i\right)}{(f_i \mu_{ci} - (f_i + r_i)\lambda_i)^3} > 0$ when $\lambda_i < \frac{f_i \mu_{ci}}{(f_i + r_i)}$, since $\mu_{ci} > 0$ and all parameters are non-negative.. Thus, W_{ci} is strictly convex w.r.t. λ_i.

Part 4: $\lambda_i W_{ci}$

$\frac{\partial \lambda_i W_{ci}}{\partial \lambda_i} = \frac{2 f_i \mu_{ci} (f_i + r_i) + \left(r_i^2 + v_i\right) \lambda_i (2 f_i \mu_{ci} - (f_i + r_i)\lambda_i)}{2(f_i \mu_{ci} - (f_i + r_i)\lambda_i)^2} > 0$ when $\lambda_i < \frac{f_i \mu_{ci}}{(f_i + r_i)}$. Thus, $\lambda_i W_{ci}$ is strictly increasing w.r.t. λ_i.

$\frac{\partial^2 \lambda_i W_{ci}}{\partial \lambda_i^2} = \frac{f_i \mu_{ci}\left(2f_i^2 + 2r_i^2 + f_i r_i (4 + \mu_{ci} r_i) + f_i \mu_{ci} v_i\right)}{(f_i \mu_{ci} - (f_i + r_i)\lambda_i)^3} > 0$ when $\lambda_i < \frac{f_i \mu_{ci}}{(f_i + r_i)}$, since $\mu_{ci} > 0$ and all parameters are non-negative.. Thus, $\lambda_i W_{ci}$ is strictly convex w.r.t. λ_i. □

Corollary 17. *$W_i = W_{pi} + W_{ci}$ and $\overline{W} = \sum_i \overline{W_i} = \sum_i \lambda_i W_i = \sum_i \lambda_i (W_{pi} + W_{ci})$ are strictly increasing and strictly convex with respect to λ_i, subject to stability conditions ($\lambda_i < \mu_{pi}$ and $\lambda_i < \frac{f_i \mu_{ci}}{(f_i + r_i)}$).*

Proof. The sum of strictly convex functions is strictly convex and the sum of strictly increasing functions is strictly increasing. □

3.3.3 Computational Insights and Examples

3.3.3.1 Impact of Breakdowns

When comparing two port-corridor paths, considering the impact of breakdowns in addition to processing rates is important in predicting expected delay. Even if Port-Corridor 2 has a faster processing rates than Port-Corridor 1, expected delays for Port-Corridor 2 can be longer than for Port-Corridor 1 if the availability of Port-Corridor 1 is sufficiently higher than Port-Corridor 2. For example, assume the following parameters: $\lambda_1 = \lambda_2 = 15$ vessels/mo., $\mu_{p1} = 25$ vessels/mo., $\mu_{c1} = 25$ vessels/mo., $f_1 = 2$ mo., $r_1 = 0.033$ mo. (1 day), $v_1 = 0.1$ mo.^{2}, $\mu_{p2} = 30$ vessels/mo., $\mu_{c2} = 30$ vessels/mo., $f_2 = 1$ mo., $r_2 = 0.25$ mo., and $v_2 = 0.1$ mo.^{2}. Even though $\mu_{p2} > \mu_{p1}$ and $\mu_{c2} > \mu_{c1}$, $W_2 = 8.58 > W_1 = 7.29$ days/vessel due to the impact of breakdowns and corridor availability ($A_1 = 0.98$ and $A_2 = 0.80$).

3.3.3.2 Variance and its Confounding Influence

When comparing two port-corridor paths, considering the impact variance is also important in predicting expected delay. Consider the case of two port-corridor networks, both with equivalent port parameters ($\lambda_1 = \lambda_2 = 15$ vessels/mo. and $\mu_{p1} = \mu_{p2} = 30$ vessels/mo.) and equivalent corridor availability ($A_1 = A_2 = 0.883$). The difference between these two port-corridor paths lies in the frequency and severity of the delays. Port-Corridor 1 has shorter failures, more often, with a mean time to failure of 1 week and a mean time to repair of 1 day. On the other hand, Port-Corridor 2 has longer failures, less often, with a mean time to failure of 12 weeks and a mean time to repair of 12 days. For both ports, we assume a fixed square coefficient of variance of repair of 1 (for a "medium level of variance" [40]), which implies $v_i = r_i^2$. The questions to be investigated are as follows: (i) do the two networks have the same waiting time performance in expectation and (ii) if not, which network has shorter expected delays?

It turns out that the two networks do not have equivalent waiting time performance. Port-Corridor 1 with shorter failures, more often, actually has less expected delays in steady state performance. This result can be seen for any stable value of $\mu_{c1} = \mu_{c2}$, which is shown in Figure 13 in which the blue function for total delay for Port-Corridor 1 is strictly below

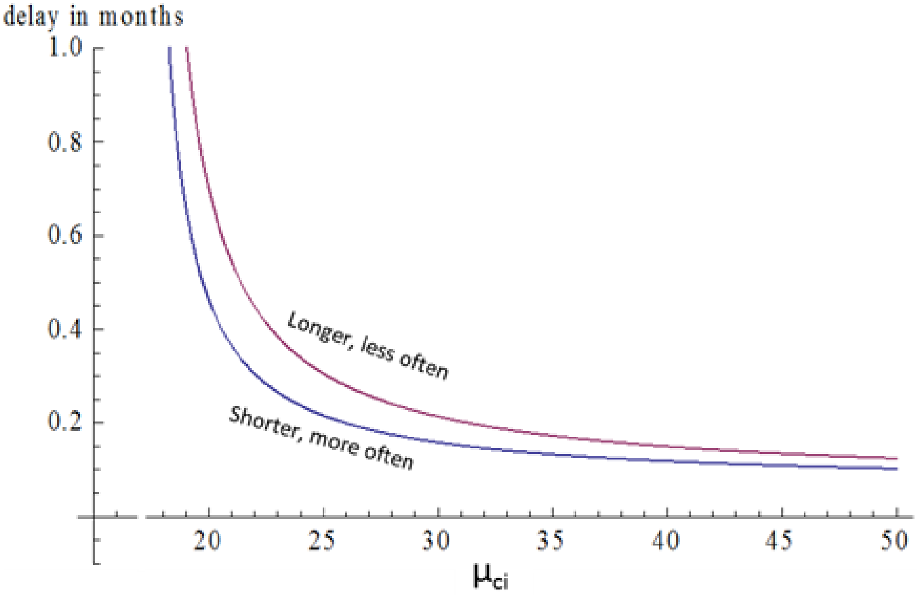


Figure 13: Total delay for two networks with the same availability but different breakdown frequency and mean time to repair as μ_{ci} increases.

the purple plot for Port-Corridor 2 ($W_{p1} + W_{c1} < W_{p2} + W_{c2}$). This example is formalized in Proposition 18.

Proposition 18. *For two stable corridors ($\lambda_i < A_i\mu_{ci}$) each with the squared coefficient of variance fixed at 1 ($v_1 = r_1^2$ and $v_2 = r_2^2$), if $\lambda_1 = \lambda_2 > 0$, $A_1 = A_2$, $\mu_{c1} = \mu_{c2}$, and $f_1 < f_2$, then $W_{c1} < W_{c2}$.*

Proof. We prove our result by showing that $W_{c2} - W_{c1} > 0$ using substitutions and the stated assumptions.

$$\begin{aligned}
W_{c2} - W_{c1} &= \frac{2(f_2 + r_2) + (r_2^2 + v_2)\lambda_2}{(2f_2\mu_{c2} - 2(f_2 + r_2)\lambda_2)} - \frac{2(f_1 + r_1) + (r_1^2 + v_1)\lambda_1}{(2f_1\mu_{c1} - 2(f_1 + r_1)\lambda_1)} \\
&= \frac{2(\frac{r_2A_2}{1-A_2} + r_2) + 2r_2^2\lambda_2}{\left(\frac{2r_2A_2\mu_{c2}}{1-A_2} - 2(\frac{r_2A_2}{1-A_2} + r_2)\lambda_2\right)} - \frac{2(\frac{r_1A_1}{1-A_1} + r_1) + 2r_1^2\lambda_1}{\left(\frac{2r_1A_1\mu_{c1}}{1-A_1} - 2(\frac{r_1A_1}{1-A_1} + r_1)\lambda_1\right)} \\
&= \frac{(1 - A_1)\,(r_2 - r_1)\,\lambda_1}{(A_1\mu_{c1} - \lambda_1)} \\
&> 0
\end{aligned}$$

□

The proof of the result centers on the "confounding influence of variance," to borrow the term from Hopp and Spearman [40]. To keep the square coefficient of variance of repair of 1, implying a medium level of variance for both paths, the variance of repair time for each path becomes r_i^2, implying that Port-Corridor 2 has larger variance of repair than Port-Corridor 1. With other factors equal (such as port conditions, A_i, and μ_{ci}), long repair times induce more variability than short ones, and this causes longer corridor delays. For certain contexts, the implication might be that controlling breakdowns through scheduled strikes or preventative maintenance could improve corridor performance, since this implies shorter more frequent planned failures instead of longer, more sporadic unplanned ones.

3.4 Incorporating Delay Modeling into Delivery Routing

Often, routing options through multiple discharge ports are available for a humanitarian operation. For example, in Syria, the main humanitarian hubs can be reached through several Lattakia, Tartous, Tripoli, and Beirut ports (Figure 14). Likewise, since multiple operations share capacity at the ports, balancing the bottlenecks and the impact of congestion of the collective routing decisions is desired. For example, operations in sub-Saharan Africa share capacity at Nouakchott, Dakar, Abidjan, Tema, Lome, and Cotonou ports (Figure 15).

In this section, we introduce a convex cost flow model that incorporates delay modeling into routing and flow decisions. Network configurations can be chosen to characterize different sets of options (e.g., minimizing just delay costs or minimizing delay costs plus fixed delivery costs). Then, we give optimality conditions for minimizing port and corridor congestion delays, for a structured network type. Finally, we evaluate a rule of thumb routing policy that might be intuitive for practitioners and make the case for an optimization approach.

3.4.1 Convex Cost Flow Model

In Theorem 16, we showed that total delay time at port and corridor stations is convex with respect to the arrival rate. We build upon this result with the convex cost network flow model in Formulation 4 that can be used to incorporate port and corridor congestion delays into a routing model. We assume independent Poisson arrival and require that on

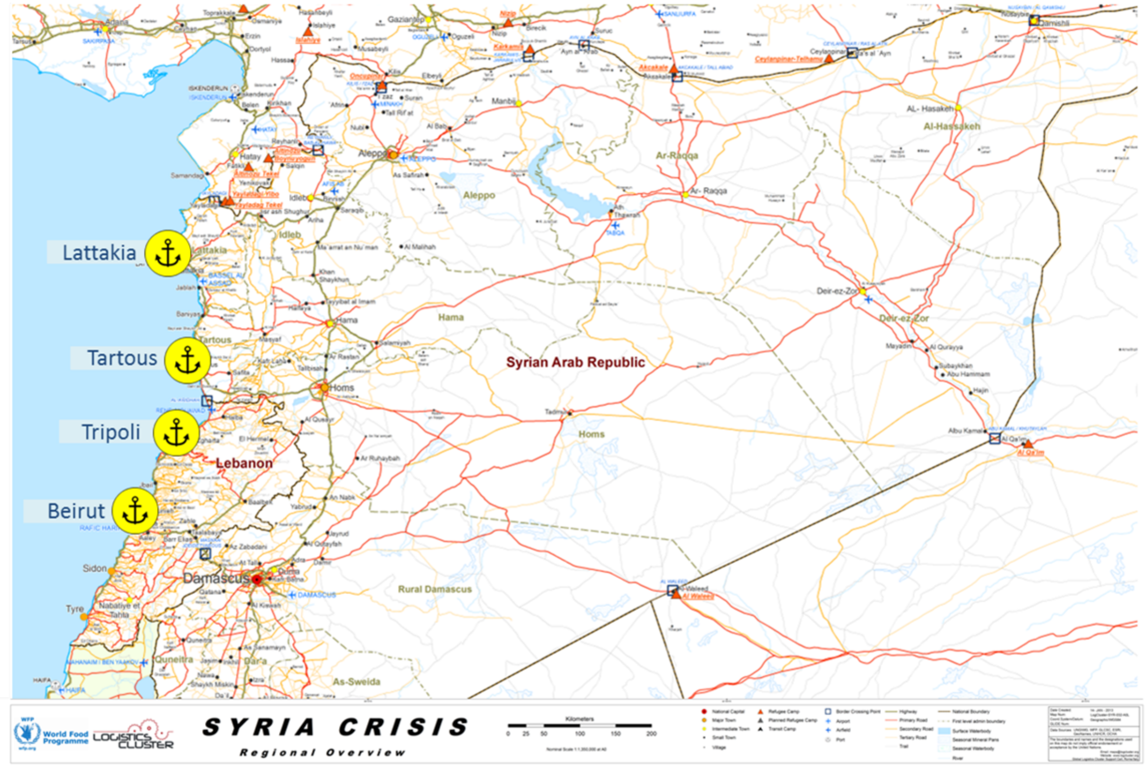


Figure 14: Main ports for Syrian humanitarian operations

any directed path in the network (i) no more than one 'corridor' station with breakdowns appear and (ii) only linear cost arcs are allowed after a 'corridor' station arc.

Formulation 4 General Convex Cost Routing Model

$$min \qquad \textstyle\sum_{(i,j)\in A} C_{ij}(\lambda_{ij}) \tag{27}$$

$$s.t. \quad \textstyle\sum_{j:(i,j)\in A} \lambda_{ij} - \sum_{j:(j,i)\in A} \lambda_{ji} = b(i) \qquad \forall i \in V \tag{28}$$

$$0 \le \lambda_{ij} \le u_{ij} \qquad \forall (i,j) \in A \tag{29}$$

In the objective function (27), each arc has a convex delay function with respect to the flow decision variable (either a convex delay cost, according to our station assumptions from Section 3.3.1, or a linear cost with respect to flow). Flow balance is ensured by Constraint (28), and Constraint (29) prevents arc capacity from being exceeded. As a convex cost flow model, the solution can be found in pseudopolynomial time, and every local minimum must be a a global minimum [20].

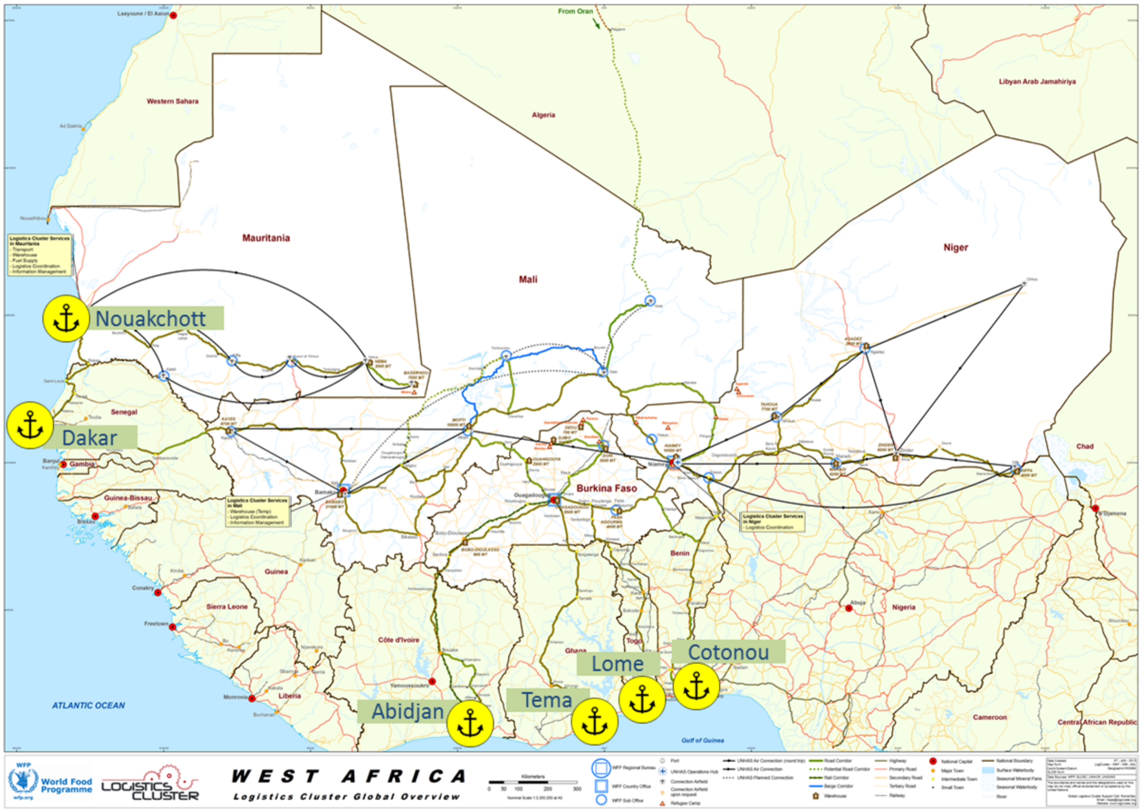


Figure 15: Main ports in West Africa for humanitarian operations

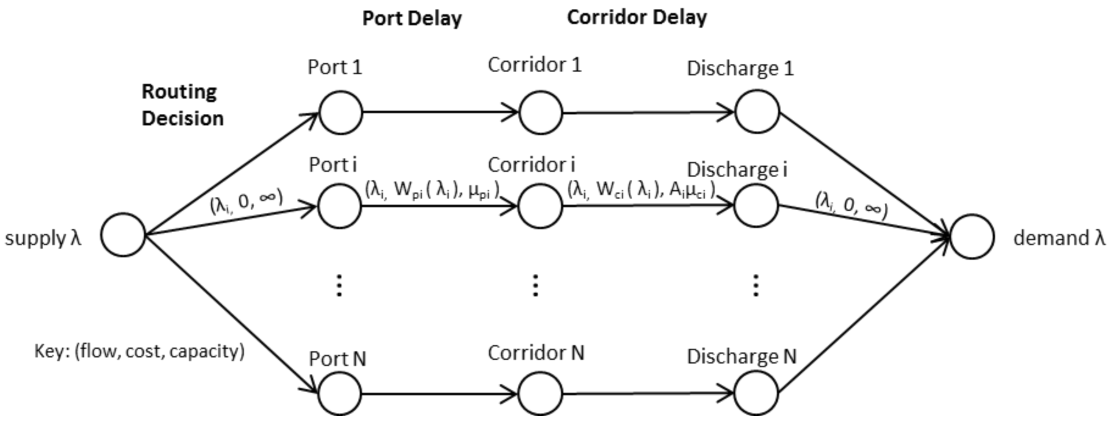


Figure 16: Port and Corridor Routing and Delay Network

Formulation 5 Port and Corridor Routing and Delay Model

$$
\begin{aligned}
&min \quad \overline{W}(\vec{\lambda}) = \textstyle\sum_i \left(\frac{1}{(\mu_{pi}-\lambda_i)} + \frac{2(f_i+r_i)+(r_i^2+v_i)\lambda_i}{(2f_i\mu_{ci}-2(f_i+r_i)\lambda_i)} \right) && (30)\\
&s.t. \qquad\qquad \textstyle\sum_i \lambda_i = \lambda && (31)\\
&\qquad\qquad\qquad \lambda_i \leq \mu_{ei} - \epsilon \qquad \forall i && (32)\\
&\qquad\qquad\qquad \lambda_i \geq 0 \qquad \forall i && (33)
\end{aligned}
$$

While different networks and sets of decisions can be modeled with this formulation (see Appendix B), one structured case that we study is the network shown in Figure 16, with the corresponding convex cost Formulation 5. This Port and Corridor Routing and Delay Model may be especially relevant for the humanitarian context where corridor delays are often concentrated in the vicinity of the port while aid awaits offtake into the corridor. The model determines routing decisions ($\lambda_i \geq 0$) that minimize the total delay in the network (Objective (30)), where a total of λ flow (at the supply node) must be routed to a demand node through one of N port-corridor paths (Constraint (31)). For each path, the *effective processing rate,* $\mu_{ei} = min(\mu_{pi},\ A_i\mu_{ci})$, is used in Constraint (32) as the maximum flow that the path can handle while maintaining our delay model stability assumptions.

3.4.2 Optimality Conditions for Minimizing Port and Corridor Delays

Focusing on our structured problem (the Port and Corridor Routing and Delay Model given in Formulation 5), we next give necessary optimality conditions relating to the partial derivatives of total wait with respect to flow. In Theorem 19, we show that for all paths that are used in an optimal solution, the partial derivative of total system wait (Objective Function 30) with respect to path flow are equivalent. The proof is based on the problem's KKT conditions.

Theorem 19. *Let $\vec{\lambda}^*$ denote an optimal solution to Formulation 5, assuming the total flow is less than the network capacity ($\lambda < \sum_i u_i$). Exactly one of the two below cases holds.*

(i) All port-corridor paths are used ($\lambda_i^* > 0, \forall i \in \{1, ..., N\}$) and the partial derivatives of total system wait w.r.t. flow are equal ($\frac{\partial \overline{W}}{\partial \lambda_1^*} = \frac{\partial \overline{W}}{\partial \lambda_2^*} = \cdots = \frac{\partial \overline{W}}{\partial \lambda_N^*}$).

(ii) Not all port-corridor paths are used ($\exists i \in \{1, ..., N\}$ s.t. $\lambda_i^* = 0$) and for the set of port-corridors path that are used, the partial derivatives of total system wait w.r.t. flow are equal ($\frac{\partial \overline{W}}{\partial \lambda_i^*} = \frac{\partial \overline{W}}{\partial \lambda_j}$ for any i and j s.t. $\lambda_i^* > 0$ and $\lambda_j^* > 0$).

Proof. The KKT Conditions for the problem are as follows:

$$\frac{\partial \overline{W}}{\partial \lambda_i} + l_0 + l_i - l_{N+i} = 0 \qquad \forall i \in \{1, ..., N\} \tag{34}$$

$$\lambda - \sum_i \lambda_i = 0 \tag{35}$$

$$u_i - \epsilon - \lambda_i \geq 0 \qquad \forall i \in \{1, ..., N\}$$

$$\lambda_i \geq 0 \qquad \forall i \in \{1, ..., N\} \tag{36}$$

$$l_j \geq 0 \qquad \forall j \in \{0, 1, ..., 2N\} \tag{37}$$

$$l_0 \left(\lambda - \sum_i \lambda_i \right) = 0 \tag{38}$$

$$l_i \left(u_i - \epsilon - \lambda_i \right) = 0 \qquad \forall i \in \{1, ..., N\} \tag{39}$$

$$l_{N+i} \lambda_i = \ \ 0 \quad \forall i \in \{1, ..., N\} \tag{40}$$

For any $\vec{\lambda}^*$ optimal solution to an instance where $\lambda < \sum_i u_i$, as per our theorem's assumptions and for both cases, we have $\overline{W}_i \to \infty$ as $\lambda_i \to u_i$. As such, for each instance $\exists \ \epsilon > 0$, s.t. $u_i - \epsilon - \lambda_i^* > 0$, and $l_i = 0, \forall i \in \{1, ..., N\}$ in order to satisfy complementary slackness in Equation (39).

For Case (i), by assumption, $\lambda_i^* > 0, \forall i \in \{1, ..., N\}$. In order to satisfy complementary slackness in Equation (40), we have $l_{N+i} = 0, \forall i \in \{1, ..., N\}$. With $l_i = l_{N+i} = 0, \forall i \in \{1, ..., N\}$, Equation (34) then reduces to $\frac{\partial \overline{W}}{\partial \lambda_1^*} = \frac{\partial \overline{W}}{\partial \lambda_2^*} = \cdots = \frac{\partial \overline{W}}{\partial \lambda_N^*} = l_0$. For Case (ii), assume

$\exists i \in \{1, ..., N\}$ s.t. $\lambda_i^* = 0$. For paths with positive flow where $\lambda_k^* > 0$, then $l_{N+k} = 0$ in order to satisfy complementary slackness in Equation (40). Let $I = \{i : \ \lambda_i^* > 0\}$. For any i and j in I, $l_i = l_j = l_{N+i} = l_{N+j} = 0$, and we see from Equation (34) that $\frac{\partial \overline{W}}{\partial \lambda_i^*} = \frac{\partial \overline{W}}{\partial \lambda_j^*} = l_0$. Clearly, Case (i) and (ii) are mutually exclusive and cover the set of possibilities for an instance. □

3.4.3 Computational Experiments

We next evaluate the performance of a potential rule of thumb policy that might be used by practitioners faced with the decision of how much of the required flow for an operation or set of operations to route through each of several port options. Here, we focus on the case of minimizing congestion delays on each port-corridor path for our structured case relevant to the humanitarian context.

3.4.3.1 Rule of Thumb Policy - Route Proportional to Each Path's Effective Processing Rate

We introduce a *Proportional to Path Effective Processing Rate* routing policy in Algorithm 3.1. In this policy, the flow allocated to each path is proportional to its *path effective processing rate*, $\mu_{ei} = min \ \{\mu_{pi}, \ A_i\mu_{ci}\}$. Practitioners might consider such a policy because it accounts for the most restricted station utilization in each path balances the flow across stations according to this metric. The policy does not take variability of repairs into account, since it is hard for practitioners to account for variability in rule of thumb policies.

For each each port-corridor path i, we consider routing in proportion to its effective processing rate. The corridor processing rate, μ_{ci}, is scaled down by the proportion of time that the corridor is available, A_i, to produce the *corridor effective processing rate*, $A_i\mu_{ci}$. Because we do not model breakdowns at the port station, the *port effective processing rate* is the same as the port processing rate, μ_{pi}. By taking the minimum of the path and effective corridor processing rates per path, the path effective processing rate incorporates the intuition that each path is as restricted as its most inefficient station. The *total effective processing rate for the network* is then $\mu_e = \sum_i \mu_{ei}$. Using the path and total effective processing rates, we consider the routing policy of setting $\lambda_i = \frac{\mu_{ei}}{\mu_e}\lambda$.

Algorithm 3.1 *Proportional to Path Effective Processing Rate* routing policy

```
1. Compute the path effective processing rate, $\mu_{ei} = min\ \{\mu_{pi},\ A_i\mu_{ci}\}$, for $i \in \{1, ..., N\}$,
   and the total effective processing rate for the network, $\mu_e = \sum_i \mu_{ei}$.
2. Assign the flow per path as $\lambda_i = \frac{\mu_{ei}}{\mu_e}\lambda$, for $i \in \{1, ..., N\}$.
```

3.4.3.2 Computational Example

Using parameters that emulate the current, ongoing humanitarian crisis in Syria, developed in consultation with expert opinion at a large humanitarian organization, we next test performance of the *Proportional to Path Effective Processing Rate* routing policy. The four main entry ports for humanitarian operations are Lattakia and Tartous in Syria and Tripoli and Beirut in Lebanon (recall Figure 14). For each, estimates for port and offtake capacity into the corridor are given in Table 3 in addition to failure and repair parameters. Due to sporadic violence in and around Homs, Tartous and Tripoli have higher mean time to repair values. For example, an attack in Homs in February 2014 disrupted the transport of humanitarian aid [12]. Though a distribution for corridor repair time is not assumed, for all paths, we set $v_i = r_i^2$, indicating a medium-level of variance where the square coefficient of variance is equal to one.

Table 3: Estimated Port-Corridor Parameters for Syria (μ_{ji} processing rates are in vessels/month; other units are as indicated)

Path	μ_{pi}	μ_{ci}	f_i	r_i	v_i	μ_{ei}
Beirut	20 v./mo.	15 v./mo.	2 mo.	1 day	(1 day)^2	14.75 v./mo.
Lattakia	30 v./mo.	41.6 v./mo.	0.5 mo.	1 day	(1 day)^2	30 v./mo.
Tartous	30 v./mo.	41.6 v./mo.	2 mo.	0.5 mo.	(0.5 mo.)^2	30 v./mo.
Tripoli	117 v./mo.	15 v./mo.	2 mo.	0.5 mo.	(0.5 mo.)^2	12 v./mo.

For $\lambda \in \{1, ..., 86\}$, just under the total network capacity of $\mu_e = 14.75 + 30 + 30 + 12 = 86.75$, we plot the optimality gap of the rule of thumb routing policy in Figure 17. For this example, the optimality gap hovers around 8-10% for most typical loads on the network (between $\lambda = 25$ and $\lambda = 84$). When total flow is lower in the network, the algorithm

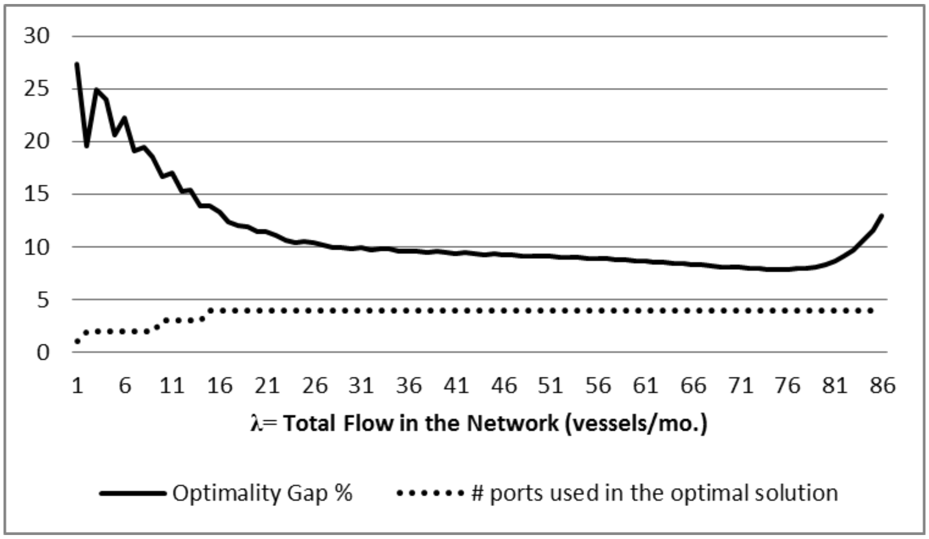


Figure 17: Performance of the *Proportional to Path Effective Processing Rate* routing policy

performs its worst. Intuition behind this result can be gained by observing that for low levels of flow, the optimal solution does not use all of the ports (see the dotted line in Figure 17) since some paths may dominate others, while the heuristic always uses all of the ports. Another general reason behind the optimality gap is that the impact of variance is not well-accounted for in the heuristic, since the corridor effective processing rate is based on f_i, r_i, and μ_{ci} but not v_i. Thus, the proportion of flow in the heuristic to Lattakia and Tartous is the same, even though Tartous is a more unreliable path, while in the optimal solution the impact of variance is accounted for and flow to Lattakia exceeds that to Tartous (e.g. as seen in Figure 18 for $\lambda = 31$, an estimated 2013 monthly total flow).

Generally, this example illustrates that while intuition and rules of thumb may be useful in practice, an optimal solution that fully factors in all considerations, including variance, may be worth investing in, especially since one can be found in pseudopolynomial time through convex cost flow modeling (e.g., using Excel or GAMS). Next, we focus on where budgets for improvement should be invested to reduce delays in the network (or alternatively make the network more capable for short-term, rapid scale-ups).

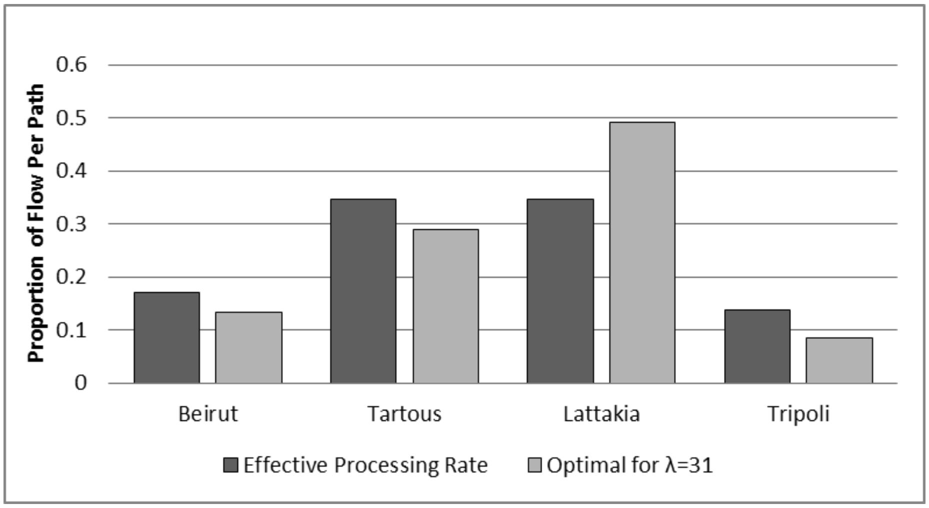


Figure 18: Proportion of flow per path: *Proportional to Path Effective Processing Rate* vs. Optimal for $\lambda = 31$, the estimated 2013 monthly total flow

3.5 Network Improvements: Investments for Decreased Congestion Delays

In this section, we allocate a limited budget for investment to improve the underlying port and corridor network. First, we characterize the monotonic impact of parametric changes on total wait and introduce stylized examples to gain insight and intuition about the problem. Second, we formulate a mathematical program that simultaneously invests a budget and routes flow optimally through the network. Third, we evaluate the performance of two potential rule of thumb policies that might be used by practitioners faced with the decision of how to allocate a limited budget for network improvements.

3.5.1 Impact of Parametric Changes

Understanding what types of investments can improve total delay in the network requires an understanding of how parametric changes impact total wait in our structured network case ($\bar{W}$ in Equation 30). For example, intuition tell us that, other factors being equal, increasing port or corridor processing rates decreases wait. It is also intuitive that having shorter failures that occur less often and with less variance in repair improves network reliability and reduces delay. In Theorem 20, we formalize this intuition by showing whether $W_i = W_{pi} + W_{ci}$ is increasing or decreasing with respect to each of the parameters that characterize port-corridor i performance, μ_{pi}, μ_{ci}, f_i, r_i, and v_i.

Theorem 20. *For a port-corridor pair i with positive flow (where $\lambda_i > 0$), the following conditions hold, assuming stability conditions for each station ($\mu_{pi} > \lambda_i$ and $f_i\mu_{ci} > (f_i + r_i)\lambda_i$).*

(i) W_i is strictly decreasing and strictly convex w.r.t. μ_{pi}.

(ii) W_i is strictly decreasing and strictly convex w.r.t. μ_{ci}.

(iii) W_i is strictly decreasing and strictly convex w.r.t. f_i when $r_i > 0$.

(iv) W_i is strictly increasing and strictly convex w.r.t. r_i.

(v) W_i is strictly increasing and linear w.r.t. v_i.

Proof. Part (i):

$\frac{\partial W_i}{\partial \mu_{pi}} = -\frac{1}{(\mu_{pi}-\lambda_i)^2} < 0$ when $\lambda_i < \mu_{pi}$, since $\mu_{pi} > 0$. Thus, W_i is strictly decreasing w.r.t. μ_{pi}.

$\frac{\partial^2 W}{\partial \mu_{pi}^2} = \frac{2}{(\mu_{pi}-\lambda_i)^3} > 0$ when $\lambda_i < \mu_{pi}$, since $\mu_{pi} > 0$. Thus, W_i is strictly convex w.r.t. μ_{pi}.

Part (ii):

$\frac{\partial W_i}{\partial \mu_{ci}} = -\frac{2f_i(2(f_i+r_i)+(r_i^2+v_i)\lambda_i)}{(2f_i\mu_{ci}-2(f_i+r_i)\lambda_i)^2} < 0$ when $f_i\mu_{ci} > (f_i + r_i)\lambda_i$, since $\mu_{ci} > 0$ and all parameters are non-negative. Thus, W_i is strictly decreasing w.r.t. μ_{ci}.

$\frac{\partial^2 W_i}{\partial \mu_{ci}^2} = \frac{f_i^2(2(f_i+r_i)+(r_i^2+v_i)\lambda_i)}{(f_i\mu_{ci}-(f_i+r_i)\lambda_i)^3} > 0$ when $f_i\mu_{ci} > (f_i + r_i)\lambda_i$, since $\mu_{ci} > 0$ and all parameters are non-negative. Thus, W_i is strictly convex w.r.t. μ_{ci}.

Part (iii):

$\frac{\partial W_i}{\partial f_i} = \frac{(r_i^2+v_i)\lambda_i^2-\mu_{ci}(2r_i+(r_i^2+v_i)\lambda_i)}{2(f_i\mu_{ci}-(f_i+r_i)\lambda_i)^2} < 0$ when $r_i > 0$ and $f_i\mu_{ci} > (f_i + r_i)\lambda_i$, since $\mu_{ci} > \lambda_i$ and all parameters are non-negative. Thus, W_i is strictly decreasing w.r.t. f_i when $r_i > 0$.

$\frac{\partial^2 W_i}{\partial f_i^2} = \frac{(\mu_{ci}-\lambda_i)(2\mu_{ci}r_i+\mu_{ci}(r_i^2+v_i)\lambda_i-(r_i^2+v_i)\lambda_i^2)}{(f_i\mu_{ci}-(f_i+r_i)\lambda_i)^3} > 0$ when $r_i > 0$ and $f_i\mu_{ci} > (f_i + r_i)\lambda_i$, since $\mu_{ci} > \lambda_i$ and all parameters are non-negative. Thus, W_i is strictly convex w.r.t. f_i when $r_i > 0$.

Part (iv):

$\frac{\partial W_i}{\partial r_i} = \frac{2f_i\mu_{ci}+2f_i\mu_{ci}r_i\lambda_i+(-r_i(2f_i+r_i)+v_i)\lambda_i^2}{2(f_i\mu_{ci}-(f_i+r_i)\lambda_i)^2} > 0$ when $f_i\mu_{ci} > (f_i + r_i)\lambda_i$, $f_i > 0$, $r_i \geq 0$, and $v_i \geq 0$. Thus, W_i is strictly increasing w.r.t. r_i.

$\frac{\partial^2 W_i}{\partial r_i^2} = \frac{\lambda_i(2f_i\mu_{ci}+f_i^2(\mu_{ci}-\lambda_i)^2+v_i\lambda_i^2)}{(f_i\mu_{ci}-(f_i+r_i)\lambda_i)^3} > 0$ when $f_i\mu_{ci} > (f_i+r_i)\lambda_i$, since $\mu_{ci} > \lambda_i$ and all parameters are non-negative. Thus, W_i is strictly convex w.r.t. r_i.

Part (v):

$\frac{\partial W_i}{\partial v_i} = \frac{\lambda_i}{(2f_i\mu_{ci}-2(f_i+r_i)\lambda_i)} > 0$ when $f_i\mu_{ci} > (f_i+r_i)\lambda_i$, since $\lambda_i > 0$. Thus, W_i is strictly increasing w.r.t. v_i.

$\frac{\partial^2 W_i}{\partial v_i^2} = 0$. Thus, W_i is linear w.r.t. v_i. □

Moving beyond the monotonic impact of parametric changes on total wait, we next explore examples to gain insight into the nature of investments in our context, such as the impact of corridor availability, diminishing rate of returns, and bottleneck shifts.

3.5.1.1 Reducing Delays by Improving Corridor Availability

Increasing time between failures (f_i) We examine three plots of total waiting time ($W_{pi} + W_{ci}$) as μ_{ci} changes, for values of the mean time to failure, f_i, in the logarithmically increasing set {1 week, 4 weeks, 16 weeks} in Figure 19. Arrivals occur at rate $\lambda_i = 15$ vessels/mo., and the port processing rate is $\mu_{pi} = 25$ vessels/mo. The mean time to repair is r_i =1 day, and we assume repair time and the square coefficient of variance of the repair time is fixed at 1. We note that as f_i increases, so does A_i, since r_i is fixed. We observe diminishing returns on improvements in total delay for increasing f_i. Also, beyond a certain μ_{ci} level, there is little impact on total delay from increasing f_i because each arriving vessel is processed more immediately upon arrival, allowing less build-up of a queue and a lesser impact on waiting time per vessel when breakdowns occurs.

Decreasing repair time (r_i) In a similar spirit to the previous analysis, in Figure 20, we examine the impact of the mean time to repair, r_i, on total waiting time. We examine three plots of total waiting time ($W_{pi} + W_{ci}$) as μ_{ci} changes, for values of the mean time to repair, r_i, in the logarithmically increasing set {6 hrs, 24 hrs, 96 hrs}. Arrivals occur at rate $\lambda_i = 15$ vessels/mo., and the port processing rate is $\mu_{pi} = 30$ vessels/mo. f_i is fixed at 1 month between failures, and the variance of repairs, v_i, is fixed at r_i^2, the value for which the square coefficient of variance of the repair time is equal to 1. We note that as r_i decreases,

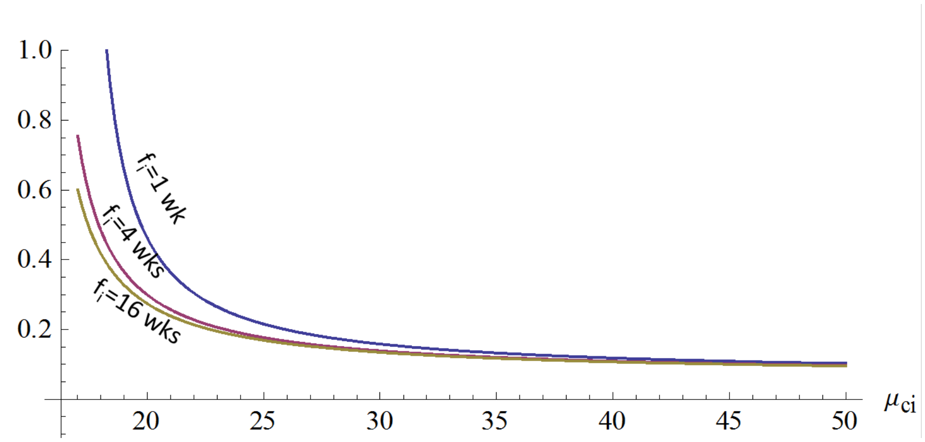


Figure 19: Total wait for different values of f_i as μ_{ci} increases. Increasing f_i improves A_i and decreases total delay.

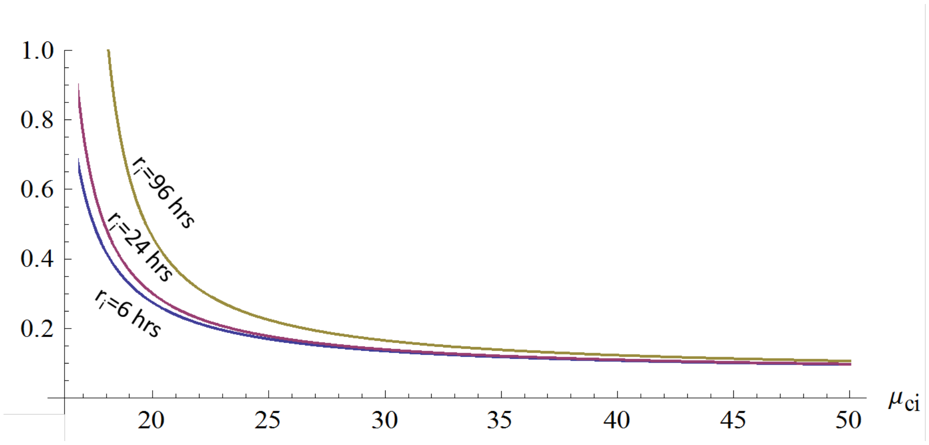


Figure 20: Total wait for different values of r_i as μ_{ci} increases. Decreasing r_i improves A_i and decreases total delay.

A_i increases, since f_i is fixed, and beyond a certain μ_{ci} level, there is little impact on total delay from decreasing r_i, again due to each vessel being processed more immediately upon arrival.

3.5.1.2 Bottleneck Shifts and Diminishing Rate of Return

We also note that while one station might be a *bottleneck,* or the station in which the most delay occurs, before investments, after investments that might change. For example, consider the chart in Figure 21, where the bottleneck station depends on the level of investment made in μ_{pi}. All parameters besides μ_{pi} are fixed ($\lambda_i = 15$ vessels/mo., $\mu_{ci} = 30$ vessel-loads/mo, $f_i = 1$ month, $r_i = 4$ days, $v_i = r_i^2$), and $W_{pi} - W_{ci}$ is plotted as μ_{pi} increases, for $\mu_{pi} > \lambda_i$.

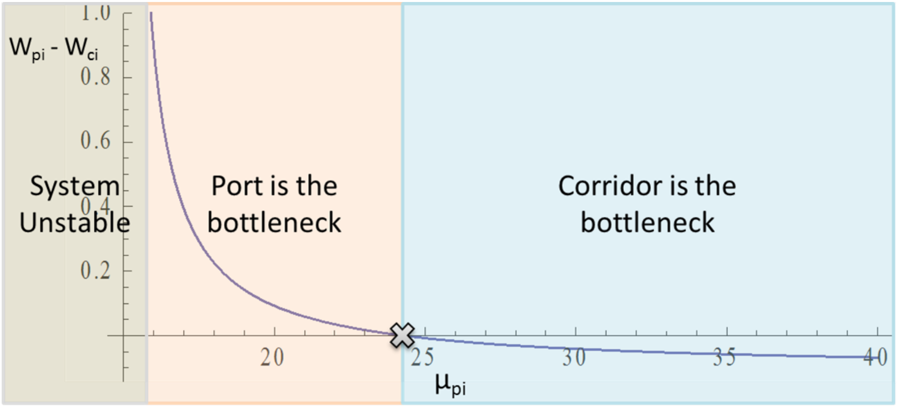


Figure 21: $W_{pi} - W_{ci}$ as μ_{pi} increases. The port is the time bottleneck for values of μ_{pi} left of the marking where the function crosses the $x - axis$, and the corridor is the bottleneck to the right.

When $W_{pi} - W_{ci}$ is above the x-axis, the port is station is the bottleneck (pictured as the orange region), and when the function is below the x-axis, the corridor is the time bottleneck (pictured as the blue region). In the plotted scenario, a port with $\mu_{pi} = 20$ vessels/mo. would be the time bottleneck in the network, but this could be shifted by increasing μ_{pi} above 24 vessels/mo. (e.g. by increasing bagging machine capacity at the port, as was done in Djibouti in 2011 [25]).

Further, throughout each of the examples and as expected due to the convexity shown in Theorem 20, a diminishing rate of return is seen in investments that are made in individual parameters. For example, in Figure 21, the first unit of increase in μ_{pi} from 20 to 21 saw a larger decrease in total wait than any other unit increase in μ_{pi} thereafter. This motivates carefully choosing how much to invest to create the largest improvements in delay in the network. We next study where to invest.

3.5.1.3 Failure Rate versus Corridor Capacity Investment at a Single Corridor Station

We present analysis for a single corridor station, where evaluating a simple expression can give insight into whether it is better to contract for additional corridor capacity or focus on reducing the frequency of failures. When resource allocations are limited to the corridor and confined to making adjustments to μ_{ci} and f_i , we analyze which parameter should be increased for a small ϵ-step to cause the largest reduction of waiting time. We assume that marginal investment costs are equivalent, allowing the results to focus on which parametric adjustment produces a steeper initial drop in waiting time.

Under these structured conditions, the following theorem gives a simple rule for determining whether it is better to focus on μ_{ci} or f_i (capacity or failures). If $(\mu_{ci} - \lambda_i) - \frac{1}{W_{ci}} > f_i$, then it is better to invest in increasing f_i than increasing μ_{ci}. On the other hand, if $(\mu_{ci} - \lambda_i) - \frac{1}{W_{ci}} < f_i$, then it is better to invest in increasing μ_{ci} than increasing f_i.

Theorem 21. *For a stable port-corridor i ($\mu_{pi} > \lambda_i$ and $f_i\mu_{ci} > (f_i + r_i)\lambda_i$), assuming $\lambda_i > 0$,μ_{pi}, r_i, and v_i cannot be changed and equivalent marginal investment costs, for at least an ϵ-step of improvement for some $\epsilon > 0$:*

(i) if $(\mu_{ci} - \lambda_i) - \frac{1}{W_{ci}} > f_i$, then it is better to invest in increasing f_i than increasing μ_{ci}

(ii) if $(\mu_{ci} - \lambda_i) - \frac{1}{W_{ci}} < f_i$, then it is better to invest in increasing μ_{ci} than increasing f_i

Proof. Case (i):

$(\mu_{ci} - \lambda_i) - \frac{1}{W_{ci}} > f_i \iff |\frac{\partial W_{ci}}{\partial \mu_{ci}}| - |\frac{\partial W_{ci}}{\partial f_i}| < 0$

Since marginal investment costs are equal, $|\frac{\partial W_{ci}}{\partial \mu_{ci}}| - |\frac{\partial W_{ci}}{\partial f_i}| < 0$ implies that $\exists \epsilon > 0$ such that the corridor wait is reduced more by marginally increasing the mean time to failure from f_i to $f_i + \epsilon$ than increasing the corridor processing rate from μ_{ci} to $\mu_{ci} + \epsilon$.

Case (ii):

$(\mu_{ci} - \lambda_i) - \frac{1}{W_{ci}} < f_i \iff |\frac{\partial W_{ci}}{\partial \mu_{ci}}| - |\frac{\partial W_{ci}}{\partial f_i}| > 0$

Since marginal investment costs are equal, $|\frac{\partial W_{ci}}{\partial \mu_{ci}}| - |\frac{\partial W_{ci}}{\partial f_i}| > 0$ implies that $\exists \epsilon > 0$ such that the corridor wait is reduced more by marginally increasing the corridor processing rate from μ_{ci} to $\mu_{ci} + \epsilon$ than increasing the mean time to failure from f_i to $f_i + \epsilon$. □

3.5.2 Simultaneously Optimizing for Network Improvements and Routing

We next model the general case where the network parameters can be improved by investing in different parts of the network. For example, purchasing bagging machines or contracting more trucking capacity can improve the underlying port and corridor service rates, respectively, and preventative maintenance or planned strikes could alter the breakdown frequency and duration and variability of repair. In Formulation 6, we present a model that simultaneously optimizes for investments in the network and routing decisions.

Formulation 6 Simultaneous Investment and Routing Model

$$\begin{aligned}
&min && \textstyle\sum_{(i,j)\in A} W_{ij}(\vec{\lambda}, \vec{\delta}) && && (41)\\
&s.t. && F'y + c'\delta \leq B && && (42)\\
& && \textstyle\sum_{j:(i,j)\in A} \lambda_{ij} - \sum_{j:(j,i)\in A} \lambda_{ji} = b(i) && \forall i \in V && (43)\\
& && 0 \leq \lambda_{ij} \leq u_{ij}(\vec{\delta}) && \forall (i,j) \in A && (44)\\
& && 0 \leq \tfrac{\delta}{U} \leq y && && (45)\\
& && y'l \leq \delta \leq d && && (46)\\
& && y \text{ binary} && && (47)
\end{aligned}$$

In the objective (41), we minimize total delay subject to routing and investment decisions ($\vec{\lambda}$ and $\vec{\delta}$, respectively). Total investments are limited by a budget in Constraint (42), with investment decisions incurring fixed and variable costs (F and c, respectively, and binary investment variables y being used to charge for fixed costs). As in Formulation 4 (and assuming the same underlying network requirements as in Section 3.4.1), flow balance and capacity constraints (43) and (44) are used for routing decisions. A distinction that is that capacity, u, is now a function of investment decisions. Constraint (45) is used to appropriately set y binary variables to indicate investments, and Constraint (46) provides upper and lower bounds on investment decisions, where lower bounds only apply in the event that an investment is made.

We continue with our structured network from Figure 16, which as we have noted may be particularly applicable to the humanitarian context. In this case, the routing decision reduces to how to split the flow among N port-corridor paths, and investment decisions are with respect to path parameters (μ_{pi}, μ_{ci}, f_i, r_i, v_i). Notation is summarized in Table 4.

Table 4: Investment Modeling Additional Notation

Notation	**Meaning**
$(c_{pi},\, c_{ci},\, c_{fi},\, c_{ri},\, c_{vi})$	Variable costs of investing in respective parameters for port-corridor pair i
$\mu_{pi} + \delta_{pi}$	Port i processing rate after investment δ_{pi} for cost $c_{pi}\delta_{pi}$
$\mu_{ci} + \delta_{ci}$	Corridor i processing rate after investment δ_{cpi} for cost $c_{ci}\delta_{ci}$
$f_i + \delta_{fi}$	Mean time to failure in Corridor i after investment δ_{fi} for cost $c_{fi}\delta_{fi}$
$r_i - \delta_{ri} \geq 0$	Mean time to repair in Corridor i after investment δ_{ri} for cost $c_{ri}\delta_{ri}$
$v_i - \delta_{vi} \geq 0$	Variance of repair in Corridor i after investment δ_{vi} for cost $c_{vi}\delta_{ri}$
B	Total budget for improvements
$(F_{pi},\, F_{ci},\, F_{fi},\, F_{ri},\, F_{vi})$	Fixed costs of investing in respective parameters for port-corridor pair i
$(y_{pi},\, y_{ci},\, y_{fi},\, y_{ri},\, y_{vi})$	Binary variable indicating whether or not investments are made in respective parameters for port-corridor pair i
U	Upper bound on the largest δ_{ji} possible for an instance (e.g. $\frac{B}{c_{min}}$ where c_{min} is the instance's cheapest unit investment cost)
$(d_{pi},\, d_{ci},\, d_{fi},\, d_{ri},\, d_{vi})$	Upper bound in improvements possible for respective parameters for port-corridor pair i
$(l_{pi},\, l_{ci},\, l_{fi},\, l_{ri},\, l_{vi})$	Lower bound required for investment (if an investment is made) for respective parameters for port-corridor pair i

The total wait across all arrivals at station i is then a function of the arrivals (λ_i) and the non-negative investment decisions $(\delta_{pi}, \delta_{ci}, \delta_{fi}, \delta_{ri}, \delta_{vi})$ and is given by Equation 48 below. A positive δ_{ji} is an improvement of a parameter (e.g., $\delta_{fi} > 0$ increases the mean time to failure, while $\delta_{ri} > 0$ decreases the mean time to repair).

Formulation 7 Structured Case: Investment and Routing Model

$$\begin{array}{rll}
min & \sum_i \overline{W}_i(\lambda_i, \delta_{pi}, \delta_{ci}, \delta_{fi}, \delta_{ri}, \delta_{vi}) & (49) \\
s.t. & \sum_i (F_{pi}y_{pi} + F_{ci}y_{ci} + F_{fi}y_{fi} + F_{ri}y_{ri} + F_{vi}y_{vi}) & \\
& + \sum_i (c_{pi}\delta_{pi} + c_{ci}\delta_{ci} + c_{fi}\delta_{fi} + c_{ri}\delta_{ri} + c_{vi}\delta_{vi}) \leq B & (50) \\
& \sum_i \lambda_i = \lambda & (51) \\
& \lambda_i - \delta_{pi} \leq \mu_{pi} - \epsilon \qquad \forall i & (52) \\
& \lambda_i(f_i + \delta_{fi} + r_i - \delta_{ri}) \leq (f_i + \delta_{fi})(\mu_{ci} + \delta_{ci}) - \epsilon \qquad \forall i & (53) \\
& r_i - \delta_{ri} \geq 0 \qquad \forall i & (54) \\
& v_i - \delta_{vi} \geq 0 \qquad \forall i & (55) \\
& \lambda_i \geq 0 \qquad \forall i & (56) \\
& 0 \leq \frac{\delta_{ji}}{U} \leq y_{ji} \qquad \forall i, \forall j \in \{p, c, f, r, v\} & (57) \\
& \delta_{ji} \leq d_{ji} \qquad \forall i, \forall j \in \{p, c, f, r, v\} & (58) \\
& \delta_{ji} \geq y_{ji} l_{ji} \qquad \forall i, \forall j \in \{p, c, f, r, v\} & (59) \\
& y_{pi}, y_{ci}, y_{fi}, y_{ri}, y_{vi} \in \{0, 1\} \qquad \forall i & (60)
\end{array}$$

$$\overline{W}_i(\lambda_i, \delta_{pi}, \delta_{ci}, \delta_{fi}, \delta_{ri}, \delta_{vi}) =$$

$$\lambda_i \left(\frac{1}{(\mu_{pi} + \delta_{pi} - \lambda_i)} + \frac{2(f_i + \delta_{fi} + r_i - \delta_{ri}) + (r_i - \delta_{ri})^2 + (v_i - \delta_{vi})\lambda_i}{(2(f_i + \delta_{fi})(\mu_{ci} + \delta_{ci}) - 2(f_i + \delta_{fi} + r_i - \delta_{ri})\lambda_i)} \right) \tag{48}$$

Our investment problem can be expressed in Formulation 7. The objective (49) minimizes the total wait as a result of routing and investment decisions. Constraint (50) ensures that investment budget B is not exceeded by summation of fixed and variable investment costs. All flow is routed in Constraint (51) and kept non-negative by Constraint (56). Port and corridor stability are ensured by Constraints (52) and (53), respectively. Constraints (54) and (55) ensure that the repair mean and variance parameters remain non-negative after investments. Investment decisions, $\vec{\delta}$, are bounded above in Constraint (58), below in Constraint (59), and binary investment indicators, $\vec{y}$, are set in Constraint (57) using a large number, U, to scale investment decisions between 0 and 1.

Unlike total wait in routing-only flow models of the previous section, Equation 48 is not convex (see Proposition 22). The proof follows from showing that $\overline{W}$ is not convex with respect to $\vec{\lambda}$, $\vec{\delta_p}$, and $\vec{\delta_c}$ using the Schur Complement of the Hessian.

Proposition 22. *The total system wait,* $\overline{W} = \sum_i \overline{W}_i(\lambda_i, \delta_{pi}, \delta_{ci}, \delta_{fi}, \delta_{ri}, \delta_{vi})$ *is not convex.*

Proof. Let H denote the $3N \times 3N$ Hessian matrix of $\overline{W}$ with respect to $\vec{\lambda}$, $\vec{\delta_p}$, and $\vec{\delta_c}$. Assume that H is divided into nine $N \times N$ matrices, H^{kl}, such that $H = \begin{bmatrix} H^{11}, & H^{12}, & H^{13} \\ H^{21}, & H^{22}, & H^{23} \\ H^{31}, & H^{32}, & H^{33} \end{bmatrix}$. Without loss of generality due to the ordering of partial derivatives with respect to $\vec{\lambda}$, $\vec{\delta_p}$, and $\vec{\delta_c}$, let $H^{11}_{ij} = \frac{\partial^2 \overline{W}}{\partial \lambda_i \partial \lambda_j}$, $H^{12}_{ij} = H^{21}_{ij} = \frac{\partial^2 \overline{W}}{\partial \lambda_i \partial \delta_{cj}}$, $H^{13}_{ij} = H^{31}_{ij} = \frac{\partial^2 \overline{W}}{\partial \lambda_i \partial \delta_{pj}}$, $H^{22}_{ij} = \frac{\partial^2 \overline{W}}{\partial \delta_{ci}^2}$, $H^{23}_{ij} = H^{32}_{ij} = \frac{\partial^2 \overline{W}}{\partial \delta_{pj} \partial \delta_{ci}} = 0$, $H^{33}_{ij} = \frac{\partial^2 \overline{W}}{\partial \lambda_i \partial \lambda_j}$, $H^{11}_{ij} = \frac{\partial^2 \overline{W}}{\partial \delta_{pi}^2}$ for $i, j \in \{1, ..., N\}$. H is a symmetric matrix.

For H^{11}, we have:

$$\frac{\partial^2 \overline{W}}{\partial \lambda_i^2} = \frac{2(\mu_{p_i} + \delta_{pi})}{(\mu_{pi} + \delta_{pi} - \lambda_i)^3} + \frac{f_i(\mu_{ci} + \delta_{ci})(2f_i^2 + 2r^2 + f_i r_i (4 + (\mu_{c_i} + \delta_{c_i}) r_i) + f_i(\mu_{c_i} + \delta_{ci}) v_i)}{(f_i(\mu_{ci} + \delta_{ci}) - (f_i + r_i)\lambda_i)^3} > 0$$

when $\lambda_i < \frac{f_i(\mu_{ci} + \delta_{pi})}{f_i + r_i}$ and $\lambda_i < \mu_{pi} + \delta_{pi}$ since μ_{ci}, $\mu_{p_i} > 0$ and all parameters are non-negative. Thus, $H^{11} > 0$, since H^{11} is a diagonal matrix ($\frac{\partial^2 \overline{W}}{\partial \lambda_i \partial \lambda_j} = 0$, when $i \neq j$) with positive entries on the diagonal.

Let $S = H^{11} - B^T C^{-1} B$ denote the Schur Complement of H^{11} in H, where $B = [H^{12},\ H^{13}]$ and $C = \begin{bmatrix} H^{22}, & H^{23} \\ H^{32}, & H^{33} \end{bmatrix}$. Indeed, C is invertible since it is a diagonal matrix ($H^{23}_{ij} = H^{32}_{ij} = 0$ and when $i \neq j$, $\frac{\partial^2 \overline{W}}{\partial \delta_{ci} \partial \delta_{cj}} = \frac{\partial^2 \overline{W}}{\partial \delta_{pi} \partial \delta_{pj}} = 0$).

Evaluating for S, we find that it is a diagonal matrix as follows:

$$S_{ij} = \begin{cases} \frac{4\lambda_i (f_i + r_i)^2}{(2(f_i + r_i) + (r_i^2 + v_i)\lambda_i)(r_i \lambda_i - f_i(\delta_{\text{ci}} - \lambda_i + \mu_{\text{ci}}))} - \frac{2\lambda_i}{(\delta_{\text{pi}} - \lambda_i + \mu_{\text{pi}})}, \text{ when } i = j \\ 0, \text{ when } i \neq j \end{cases}$$

and we see that $\frac{4\lambda_i (f_i + r_i)^2}{(2(f_i + r_i) + (r_i^2 + v_i)\lambda_i)(r_i \lambda_i - f_i(\delta_{\text{ci}} - \lambda_i + \mu_{\text{ci}}))} - \frac{2\lambda_i}{(\delta_{\text{pi}} - \lambda_i + \mu_{\text{pi}})} < 0$ when $\lambda_i < \frac{f_i(\mu_{ci} + \delta_{pi})}{(f_i + r_{)i}}$ and $\lambda_i < \mu_{pi} + \delta_{pi}$, which implies $S < 0$.

Next, we invoke a proposition from A.5.5 in [20]: if $H^{11} > 0$, then $H \geq 0$ if and only if $S \geq 0$. We have $H^{11} > 0$ and $S < 0$, thus H is not positive semi-definite. Concluding the proof, $\overline{W}$ is then not convex with respect to $\vec{\lambda}$, $\vec{\delta_p}$, and $\vec{\delta_c}$, and therefore $\overline{W}$ is not convex. □

The implication of a non-convex objective function is that local optima are not guaranteed to be globally optimal and more general nonlinear programming techniques are required to solve Formulation 7. Further, Constraint (53) is nonlinear, and there are binary variables in the problem.

However, despite these characteristics, which can be a challenge in nonlinear programming, we find that optimal solutions for this structured problem can be generated for certain problem sizes. For the UN World Food Programme operation in Syria, four main discharge ports are used (Beirut, Lattakis, Tartous, and Tripoli, as in Figure 14). In our computational testing using the free NEOS server its Baron global optimization license [54], the instances evaluated for $N = 4$ for Formulation 7 could be solved within 0.1% of optimality in under 10 seconds.

Most decisions are of this scale in the humanitarian context, with N usually ranging from 1 to 4 ports or, in the case of some operations in Western Africa, from 5 to 7 ports (as in Figure 15). For this problem size range, we found the following computational times for solutions within 0.1% of optimality using Baron on NEOS: under 10 seconds for $N \in [1, 4]$ (5-20 binary variables), under 5 minutes for $N = 5$ (25 binary variables), and under 90 minutes for $N = 6$ (30 binary variables). For $N = 7$ (35 binary variables), the Baron solver ran for 8 hours, the NEOS limit, without identifying a solution meeting the stopping criteria. We next compare solutions using the optimal approach to rules of thumb and identify important factors that might be missing in practical decision making currently.

3.5.3 Computational Experiments

For the structured case of Formulation 7, we evaluate the performance of two potential rule of thumb policies that might be used by practitioners faced with the decision of how to allocate a limited budget for network improvements. The two policies contrast the approach of focusing efforts on the single, biggest bottleneck versus spreading investments across all of the stations. Here, we focus on the case of minimizing congestion delays on each port-corridor path by making both routing and investment decisions (solving Formulation 7).

3.5.3.1 Rule of Thumb Policies

Both policies are initialized by finding the optimal flow before any investments are made (i.e. the $\vec{\lambda}^*$ that minimizes total wait in Formulation 5) and calculating total wait at each station ($\lambda_i^* W_{pi}$ and $\lambda_i^* W_{ci}$ for all paths).

All to the Bottleneck In the *All to the Bottleneck* policy (Algorithm 3.2), the station (port or corridor) with the highest expected total wait is identified and as much of the budget as is possible to be spent at this bottleneck station is invested. If the bottleneck station is a port, and if there is enough budget to cover the fixed investment cost, F_{pi}, then the budget is invested up to the upper bound on δ_{pi}. If the bottleneck station is a corridor, then multiple parameters are candidates for investment (the processing rate, mean time to failure, mean time to repair, and variance of repair time). In this case, the policy chooses the parameter with the highest reduction in wait per dollar invested, assuming the biggest investment step possible is taken in each possible direction. If budget is leftover at a corridor station after making a corridor investment (e.g. if \$100 is leftover after setting $\delta_{ri} = r_i$), then again the budget is invested according to the parameter with the highest reduction in wait per dollar. Finally, if budget is still leftover at a station due to the upper and lower bounds and fixed costs, then proceed to station with the next highest wait, and repeat. Once all of the budget has been allocated, the flow after investments is re-optimized.

Divide Proportionally Among Stations In the *Divide Proportionally* policy (Algorithm 3.3), the total budget is split among each station in proportion to its total wait before investment ($\frac{\lambda_i^* W_{pi}}{\bar{W}} B$ for each port and $\frac{\lambda_i^* W_{ci}}{\bar{W}} B$ for each corridor). At each port station, if there is enough budget to cover the fixed investment cost, F_{pi}, and lower bound on investing, l_{pi}, the allocated budget is invested up to the upper bound, d_{pi}. At each corridor station, as in the *All to the Bottleneck* policy, the allocated budget is invested in the parameter with the highest reduction in wait per dollar invested, assuming the biggest investment step possible is taken in each possible direction, and the process is repeated until investments are no longer possible at the station.

Algorithm 3.2 *All to the Bottleneck* Investment Policy

```
1. Initialize by finding the $\vec{\lambda}^*$ that minimizes total wait in Formulation 5, and calculate total wait at each station ($\lambda_i^* W_{pi}$ and $\lambda_i^* W_{ci}$ $\forall i \in \{1, ..., N\}$). Initialize investment variables $\vec{\delta} = \vec{0}$ and $\vec{y} = \vec{0}$. Let $List$=$\{\lambda_1^* W_{p1}, \lambda_1^* W_{c1}, ...., \lambda_N^* W_{pN}, \lambda_N^* W_{cN}\}$ and remaining budget, $b = B$.

2. Evaluate $argmax\{List\}$ to determine the bottleneck station, $pi$ or $ci$, with the highest expected total wait.

3. While $List$ is non-empty and $b > 0$:

    (a) If the bottleneck station is a port and if $b > F_{pi}(1 - y_{pi}) + l_{pi}c_{pi}(1 - y_{pi})$, then the increase to port investment is $\delta'_{pi} = min\left(\frac{b-F_{pi}}{c_{pi}},\ \frac{b-F_{pi}(1-y_{pi})}{c_{pi}},\ d_{pi}\right)$, and $\delta_{pi} = \delta_{pi} + \delta'_{pi}$. Decrease the budget, $b = b - F_{pi}(1 - y_{pi}) - \delta'_{pi}c_{pi}$. Remove $\lambda_i^* W_{pi}$ from $List$, update $y_{pi} = 1$, and return to Step 2.
    (b) If the bottleneck station is a corridor, then initialize $potentialInvestments$= $\{c,\ f,\ r,\ v\}$, and the algorithm proceeds as follows:
        i. While $potentialInvestments$ is non-empty and $b > 0$:
            A. For $j \in potentialInvestments$, define potential step for investment, $\delta'_{ji}$, and add a corresponding element in $ROIlist$ according to the following functions for $ROI_j$.
                • For $j \in \{c,\ f\}$, where $e_c = (0, 1, 0, 0, 0)^T$ and $e_f = (0, 0, 1, 0, 0)^T$,
                  $\delta'_{ji} = \begin{cases} min\left(\frac{b-F_{ji}}{c_{ji}},\ \frac{b-F_{ji}(1-y_{ji})}{c_{ji}},\ d_{ji}\right), & if\ b > F_{ji}(1 - y_{ji}) + l_{ji}c_{ji}(1 - y_{ji}) \\ 0, & otherwise \end{cases}$
                  $ROI_j = \begin{cases} \frac{\bar{W}(\vec{\lambda^*},\ \vec{\delta}) - \bar{W}(\vec{\lambda^*},\ \vec{\delta}+e_j\delta'_{ji})}{\delta'_{ci}}, & if\ \delta'_{ji} > 0 \\ 0, & if\ \delta'_{ji} = 0 \end{cases}$
                • For $j \in \{r,\ v\}$, where $e_r = (0, 0, 0, 1, 0)^T$ and $e_v = (0, 0, 0, 0, 1)^T$,
                  $\delta'_{ji} = \begin{cases} min\left(\frac{b-F_{ji}}{c_{ji}},\ \frac{b-F_{ji}(1-y_{ji})}{c_{ji}},\ d_{ji},\ j_i\right), if\ b > F_{ji}(1 - y_{ji}) + l_{ji}c_{ji}(1 - y_{ji}) \\ 0,\ otherwise \end{cases}$
                  $ROI_j = \begin{cases} \frac{\bar{W}(\vec{\lambda^*},\ \vec{\delta}) - \bar{W}(\vec{\lambda^*},\ \vec{\delta}-e_j\delta'_{ji})}{\delta'_{ri}}, & if\ \delta'_{ji} > 0 \mathrm{B} \\ 0, & if\ \delta'_{ji} = 0 \end{cases}$
            B. For $j \in potentialInvestments$, if $ROI_j = 0$, remove $j$ from $potentialInvestments.$
            C. Evaluate $j = argmax\{ROIlist\}$ to identify the next investment. Remove $j$ from $potentialInvestments$. Record the investment step, $\delta_{ji} = \delta_{ji} + \delta'_{ji}$; update the budget, $b = b - F_{ji}(1 - y_{ji}) - \delta'_{ji}c_{ji}$; and update $y_{ji} = 1$.
        ii. Remove $\lambda_i^* W_{ci}$ from $List$ and return to Step 2.

4. Reoptimize flow by solving Formulation 5 with updated model parameters according to investment decisions $\vec{\delta}$ ($\mu_{pi} = \mu_{pi} + \delta_{pi}$, $\mu_{ci} = \mu_{ci} + \delta_{ci}$, $f_i = f_i + \delta_{fi}$, $r_i = r_i - \delta_{ri}$, and $v_i = v_i - \delta_{vi}$) and solve for $\vec{\lambda}$ to minimize the total wait.
```

Algorithm 3.3 *Divide Proportionally* Investment Policy

```
1. Initialize by finding the $\vec{\lambda}^*$ that minimizes total wait in Formulation 5, and calculate total wait at each station ($\lambda_i^* W_{pi}$ and $\lambda_i^* W_{ci}$ $\forall i \in \{1, ..., N\}$) and in the network ($\bar{W} = \sum_i \lambda_i^*(W_{pi} + W_{ci})$). Initialize investment variables $\vec{\delta} = \vec{0}$ and $\vec{y} = \vec{0}$.
2. For $i \in N$, $\delta_{pi} = \begin{cases} min\left(\frac{\frac{\lambda_i^* W_{pi}}{\bar{W}}B - F_{pi}}{c_{ji}},\ d_{pi}\right), & if\ \frac{\lambda_i^* W_{pi}}{\bar{W}}B > F_{pi} + l_{pi}c_{pi} \\ 0, & otherwise \end{cases}$
    (a) Set remaining budget: $b_{port,i} = \begin{cases} \frac{\lambda_i^* W_{pi}}{\bar{W}}B, & if\ \delta_{pi} = 0 \\ \frac{\lambda_i^* W_{pi}}{\bar{W}}B - F_{pi} - \delta_{pi}c_{pi}, & otherwise \end{cases}$
    (b) Update the binary investment variable: $y_{pi} = \begin{cases} 1, & if\ \delta_{pi} > 0 \\ 0, & if\ \delta_{pi} = 0 \end{cases}$
3. For $i \in N$, allocate $\frac{\lambda_i^* W_{ci}}{\bar{W}}B$ to each corridor according to Algorithm 3.2 in Step 3.b.i (according to the best return on investment) to produce ($\delta_{ci}$, $\delta_{fi}$, $\delta_{ri}$, $\delta_{vi}$) and ($y_{ci}$, $y_{fi}$, $y_{ri}$, $y_{vi}$).
    (a) Set remaining budget: $b_{cor,i} = \begin{cases} \frac{\lambda_i^* W_{pi}}{\bar{W}}B, & if\ \delta_{ci} = \delta_{fi} = \delta_{ri} = \delta_{vi} = 0 \\ \frac{\lambda_i^* W_{pi}}{\bar{W}}B - \sum_j (F_{ji}y_{ji} + \delta_{ji}c_{ji}), & otherwise \end{cases}$
4. Pool the remaining budget, $b = \sum_i (b_{port,i} + b_{cor,i})$, and let *List*=$\{\bar{W}_{p1}(\vec{\lambda}, \vec{\delta})$, $\bar{W}_{c1}(\vec{\lambda}, \vec{\delta})$,..., $\bar{W}_{pN}(\vec{\lambda}, \vec{\delta})$,$\bar{W}_{cN}(\vec{\lambda}, \vec{\delta})\}$. Then, initialize the *All to the Bottleneck* investment policy to invest $b$ in the bottleneck station(s) until funds run out (Steps 2-4 in Algorithm 3.2).
```

Due to the investment upper and lower bounds and the fixed costs, budget may be leftover at one or more stations after the above steps take place. In this case, all of the leftover budget is pooled together and spent at the bottleneck station(s) according to the *All to the Bottleneck* policy. Once all of the budget has been allocated, the flow after investments is re-optimized.

3.5.3.2 Computational Example Parameters

Continuing with the example from Section 3.4.3.2, we next test performance of the *All to the Bottleneck* and *Divide Proportionally* policies. Example variable and fixed costs of investments are given in Table 5, and example upper and lower bounds on investment are given in Table 6.

Table 5: Estimated Investment Costs in thousands of USD

Path	c_{pi}	c_{ci}	c_{fi}	c_{ri}	c_{vi}	F_{pi}	F_{ci}	F_{fi}	F_{ri}	F_{vi}
Beirut	5	10	100	600	500	20	10	40	40	40
Lattakia	5	10	200	600	500	20	10	40	40	40
Tartous	5	10	100	600	500	20	10	40	40	40
Tripoli	5	10	100	600	500	20	10	40	40	40

Table 6: Upper and Lower Bounds (units correspond to those in Table 3, "-" denotes no upper bound is defined, so one can be set arbitrarily high in the formulation)

Path	u_{pi}	u_{ci}	u_{fi}	u_{ri}	u_{vi}	l_{pi}	l_{ci}	l_{fi}	l_{ri}	l_{vi}
Beirut	5	6	-	-	-	1	0	0	0	0
Lattakia	5	10	-	-	-	1	0	0	0	0
Tartous	5	10	-	-	-	1	0	0	0	0
Tripoli	5	6	-	-	-	1	0	0	0	0

3.5.3.3 Results as the Budget Increases for a Fixed Flow Level

For $B \in \{100,\ 200, ...,\ 1000\}$ and $\lambda = 69$, an estimate of the 2014 monthly flow, we plot the total wait for the two rule of thumb policies and the optimal solution in Figures 22 for the case without fixed costs or bounds. Here, we see that while both rules of thumb perform increasingly worse relative to the optimal solution as the budget increases, the *All to the Bottleneck* policy performs much worse, with savings leveling off due to diminishing returns on investment in the bottleneck station receiving all of the investment.

With fixed costs and bounds, the story is slightly less clear-cut, as can be seen in Figure 23. Here, the performance of the *All to the Bottleneck* policy surpasses that of the *Divide Proportionately* policy at $B = 700$, in this particular case because of a shift to investing in a new station after the bounds are exhausted at the previous bottleneck. As is expected, the overall saving achieved with the budget are less due to the fixed cost and bounds on investment. The take-away here is that in all cases, the optimal solution performs much better than the heuristics (with optimality gap at best over 15% and at worst over 400%) and that due to the combinatorial nature of the problem with fixed costs and bounds, intuition for what type of investment policy to use may not be clear since the optimality gap plots for the two heuristics cut across each other more than once in Figure 23.

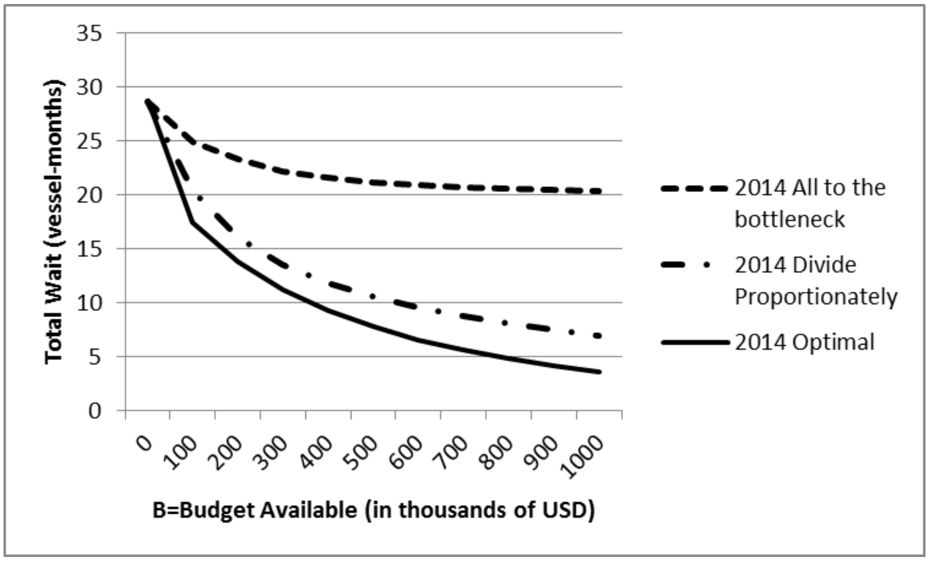


Figure 22: Total Wait as the budget increases under different policies ($\lambda = 69$ and no fixed costs or bounds are used in this experiment)

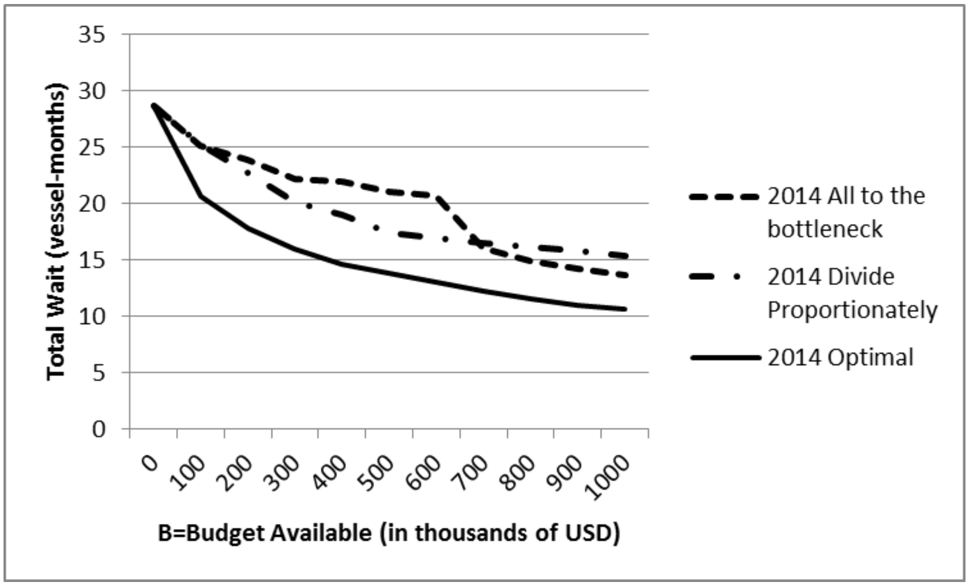


Figure 23: Total Wait as the budget increases under different policies ($\lambda = 69$ and with fixed costs and bounds in this experiment)

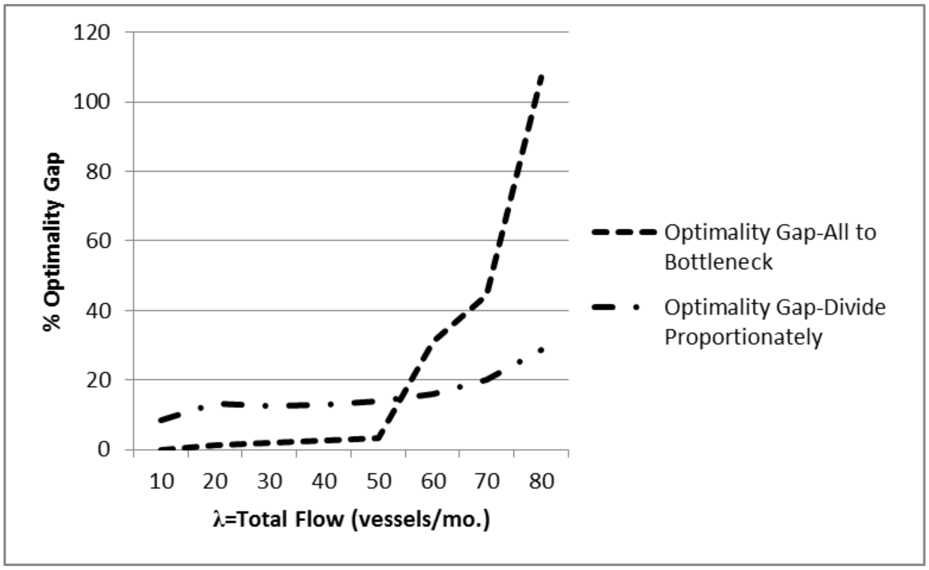


Figure 24: Optimality Gap Percentage as Flow in the Network Increases ($B = 100$; without fixed costs or bounds)

3.5.3.4 Results as Total Flow Increases for a Fixed Budget

For $\lambda \in \{10,\ 20, ..., 80\}$, under the total network capacity before investments of $\mu_e = 86.75$, we plot the optimality gap of the rule of thumb routing policies in Figures 24 and 25, without and with fixed costs and bounds, respectively. In both, we see that the *Divide Proportionately* policy performs more evenly than the *All to the Bottleneck* policy (not as well in the best case as *All to the Bottleneck* but also not as bad in the worst case). This makes sense as the *Divide Proportionately* policy takes a balanced investment approach across all stations, minimizing the maximum wait across the network stations. While the optimality gap of the *All to the Bottleneck* policy is under 5% for lower levels of flow and without fixed costs and bounds, for cases more realistic for a potential investment scenario (i.e. higher flow and with fixed costs and bounds), the optimality gap is 12% at best and 60% at worst, again motivating the usage of an optimization model. For both experiments, we see that the optimality gap worsens as total flow increases. The intuition behind this observation is that as the network approaches capacity, deviations from optimality are more costly, which we explore in more detail next.

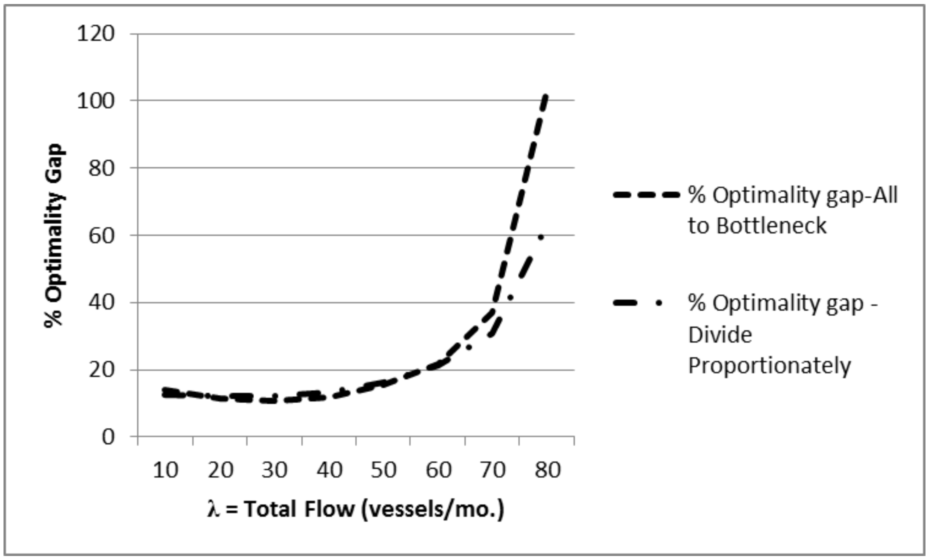


Figure 25: Optimality Gap Percentage as Flow in the Network Increases ($B = 180$; with fixed costs and bounds)

Deviations from Optimality are More Costly when the Network is Near Capacity

We next explicitly examine the assertion that as the network approaches capacity, deviations from optimality of improvement decisions are more costly. For the case with fixed costs and bounds and budget $B = 180$, the maximum flow that the network can handle under optimal investments is just over $\lambda = 102$. For this instance at $\lambda = 102$ and $B = 180$, the optimal investments are $\delta_{p,Tartous} = 3.217$, $\delta_{p,Lattakia} = 5$, $\delta_{c,Beirut} = 5.267$, and $\delta_{c,Tripoli} = 2.616$. In Figure 26, we plot the total wait where $\delta_{p,Tartous} = 3.217+\epsilon$, $\delta_{p,Lattakia} = 5-\epsilon$, $\delta_{c,Beirut} = 5.267 - \epsilon$, and $\delta_{c,Tripoli} = 2.616 + \epsilon$. Indeed, the total wait more than doubles compared to the optimal solution's total wait when $\epsilon = 0.04$, emphasizing the importance of reaching an optimal solution when making investments in a congested network.

3.6 Conclusion and Future Directions

Humanitarian transport networks are prone to congestion and disruptions due to the context in which they operate. We addressed three practical topics in this chapter: (i) congestion modeling including breakdowns, (ii) optimal routing to reduce the severity of delays, and (iii) improving network performance through resource allocation towards improving capacity or reducing the impact of breakdowns. Throughout, we compare solutions using the optimal

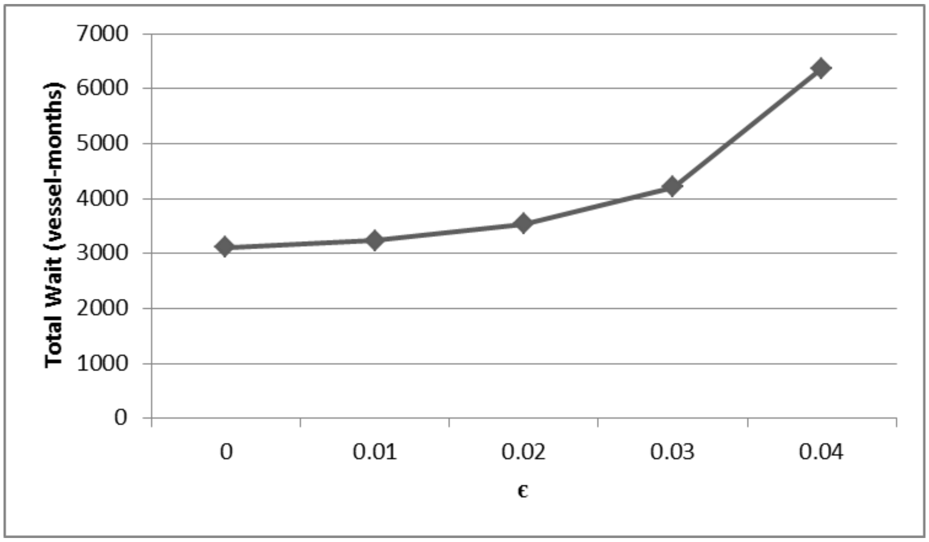


Figure 26: Total wait as ϵ increases where $\delta_{p,Tartous} = 3.217 + \epsilon$, $\delta_{p,Lattakia} = 5 - \epsilon$, $\delta_{c,Beirut} = 5.267 - \epsilon$, and $\delta_{c,Tripoli} = 2.616 + \epsilon$ ($\lambda = 102$, $B = 180$, and with fixed costs and bounds)

approach to rules of thumb and identify important factors that might be missing in practical decision making currently, such as appropriately accounting for the impact of variance of breakdown length. Moreover, the instances evaluated could be solved quickly and with open-source optimization resources for most realistic problem sizes (with the demo solvers included with GAMS for the convex cost routing models and through the NEOS Server using BARON [54] for the non-convex, binary NLP investment problem).

We first introduced a queuing-based model with stochastic breakdowns for predicting congestion in ports and corridors and obtained closed-form expressions for expected waiting time in the system. Then, we showed that these waiting time expressions were convex with respect to the arrival rate. Finally, we provided computational-based insights into congestion and highlighted that shorter breakdowns, less often can be preferable to longer breakdowns, less often due to the influence of variance of repair time in the network.

We next introduced a convex cost flow model with the objective to minimize total congestion delay. For a special structured case relevant to the humanitarian context, we showed the optimality condition that for all paths that are used in a optimal solution, the partial derivative of total wait with respect to path flow are equivalent. We evaluated the performance of a potential rule of thumb policy that might be used by practitioners to route flow (*Proportional to Path Effective Processing Rate*) in computational experiments for the structure case. We observed that the optimal solution may not use every path in a network and is better able to account for variability in the network than the rule of thumb policy.

Finally, we characterized the parametric relationships that lead to decreased total waiting time (such as increasing capacity or increasing the time between breakdowns), and we formulated a mathematical program that simultaneously invests a budget and routes flow optimally through the network. We evaluated the performance of two potential rule of thumb policies that might be used by practitioners faced with the decision of how to allocate a limited budget for network improvements (*All to the Bottleneck* and *Divide Proportionally*) for a structured case. In these computational experiments, we saw that the optimal solution performed much better than the heuristics and that due to the combinatorial nature of the problem with fixed costs and bounds, intuition for what type of investment policy to use may not be clear. We also illustrated that deviations from optimality are more costly when the network is near capacity – precisely the case when investments in the network may be most warranted.

Future research directions include incorporating congestion into other humanitarian supply chain frameworks, including those for advanced purchasing and inventory stocking levels. For the non-convex investment problem, specialized algorithms could be investigated for larger problem sizes, which may be appropriate if the approach were to be adapted for private-sector applications. Finally, different vessel sizes and cargo types (bulk vs. bagged and charter vs. liner) could be explicitly modeled and incorporated into the congestion model.

Chapter IV

A CASE STUDY ON THE IMPLEMENTATION OF SUPPLY CHAIN KEY PERFORMANCE INDICATORS AT A LARGE HUMANITARIAN ORGANIZATION

4.1 Introduction

In this chapter, we present a case study on the adoption of supply chain key performance indicators (SC KPIs) at a large humanitarian organization using action research methodology. We examine what is necessary for an implementation of KPIs for the supply chain and the challenges of performance measurement at a humanitarian institution, highlighting differences and similarities with the private sector throughout.

While performance measurement in the private sector has been used to promote efficiency in commercial organizations for decades (e.g., [43, 65]), its usage in the humanitarian sector is still relatively new. However, the recognition that supply chain performance measurement is applicable to humanitarian operations is seeing increased recognition by humanitarian organizations [66], and donors are pressing for increased operational efficiency and effectiveness [60, 79].

Despite the lack of systematic implementations of KPIs at most humanitarian organizations, several organizations engage in ad hoc reporting on KPIs at various operational levels (e.g., UNICEF tracks KPIs for ready-to-use therapeutic food [44]). The idea of performance measurement and 'proving efficiency and effectiveness' is becoming more and more demanded from humanitarian organizations [49, 60, 78]. Yet, the needed mindset [30], training [78], and infrastructure [41] is still lagging in the sector.

The research questions we investigate are the following:

1. What phases are necessary in a full implementation of KPIs for the supply chain at a humanitarian organization?

2. How can a large performance measurement initiative at a humanitarian organization

find its fit within corporate strategy, cultivate necessary mindset changes, and overcome underlying organizational barriers?

3. Do existing SC KPI approaches need adaptation for the humanitarian context? In particular,

    (a) How do implementation constraints in the humanitarian sector impact the KPIs that are developed and how does this compare with the private sector?

    (b) Can commercial supply chain KPI frameworks be adapted for humanitarian organizations or is a new framework necessary?

Systematically implementing supply chain KPIs across an entire organization, rather than on an ad hoc basis, involves ensuring the right data foundations; navigating strategic, cultural and organizational challenges; developing relevant and measurable strategic indicators; creating a means for standardized and regular dissemination throughout the organization (e.g., using an interactive dashboard); and committing to the training, tracking, and follow-through needed to achieve performance improvement. It is an extremely large and challenging undertaking that should be planned for and handled in a thoughtful and informed way.

The payoff from doing so promises to be significant. Improved performance of the humanitarian supply chain can lead to increases in the number of beneficiaries served and improvement in the quality and reliability of service, leading to more effective operations.

In the humanitarian sector, supply chain KPIs can also be used to create a more meaningful dialogue around the effectiveness of operations. Equipped with a better understanding that one way to improve effectiveness is to conduct necessary operations efficiently, donors may be more likely to invest in “smart overhead” [33]. Such an approach advocates for increased operational overhead in system-improving investments when it can result in better overall efficiency, an approach that is often overlooked by donors [85].

This chapter has the following structure. Section 4.2 presents the research methodology. Then, in Section 4.3, we frame our contributions in the context of relevant literature regarding performance measurement and the humanitarian supply chain. Section 4.4 details our case study on a systematically-implemented supply chain performance measurement initiative at a large humanitarian organization, with subsections addressing our research questions. Finally, conclusions are given in Section 4.5, and research limitations and future directions are discussed.

4.2 Research Approach

As a detailed study on phenomena at one organization addressing a set of research questions, our work on the SC KPI project fits into single-case framework according to Yin [91]. Further, with one of the authors working as a consultant and research affiliate for the organization on the project of study, the case further adopts the action research methodology according to [21, 27, 84]. Action research is summarized by Braz et al [21] as (i) work investigating more than just actions, (ii) in a participatory fashion, (iii) that occurs at the same time as the action, and (iv) is the sequence of events and approaches used to solve problems.

Serving in various roles for the studied SC KPI project for over three years, one of the authors participated in and witnessed many detailed aspects of the project over its history, including being involved in the creation of over 150 project documents and stakeholder presentations. Activities also included collaborating with the private industry experts advising and financing the project, leading the efforts to synthesize inputs and requirements for the framework and metrics and leading the business requirements for the development of the dashboard. Participation in all of these project phases enhanced and fit the action research, single case approach undertaken.

4.3 Literature Review and Contributions

Our research fits within the cross-section of performance measurement and action research case studies in the academic literature. Lohman [47] presents an action research case based on the development and dashboard implementation of performance measurement at Nike, focusing on incorporating existing company metrics into a KPI campaign, while Wouters [88] addresses characteristics and managerial implications of processes for the design and implementation of performance measurement systems, bringing together supply chain and operations management concepts. Our action research case addresses performance measurement in the humanitarian supply chain, a scope not shared by any existing papers in the literature to the best of our knowledge. We next frame our areas of contribution within the literature.

We address the complex process of large-scale implementation of SC KPIs at a humanitarian organization. Overall, usage of performance measurement in the humanitarian context is still relatively new, as outlined by Adibi and Klumpp [3] in their literature overview of the area. Other papers present conceptual examples [13, 30, 71] in the humanitarian context, but none found cover the actual project phases of officially implementing the concepts, from project buy-in and initialization through metric development and then piloting and lasting adoption of supply chain performance measurement at a humanitarian organization. We generalize the phases necessary for a full implementation of KPIs for the supply chain at a humanitarian organization.

In the private-sector context, performance measurement has been used to promote efficiency and effectiveness in private sector organizations for decades (e.g., [43, 65]), and there is a vast literature spanning the subject (see [10, 37, 52] and the references therein). The authors in [19, 47, 89] cover details of a designing, implementing, using and continuously updating performance measurement systems. We specifically address the following areas where the humanitarian context may present additional challenges: finding a fit within corporate strategy, addressing culture and mindset changes, and overcoming inherent structural barriers.

In the humanitarian context, existing literature introduces conceptual or example adaptations of SC KPI approaches [30, 13, 70, 71, 87]. Davidson [30] develops a framework, metrics, and scorecards for how to measure supply chain performance of relief operations for the logistics department of the International Federation of Red Cross and Red Crescent Societies. Beamon and Balcik [13] also develop a framework for performance measurement in the relief sector and note that performance metrics and measurement systems have not been widely developed and systematically implemented in the humanitarian organizations "due to the difficulties associated with measuring program outcomes and impacts in humanitarian relief." This research highlights areas where proposed metrics from the literature cannot be measured systematically in practice due to implementation constraints. We examine how this impacts the adaptation of existing private sector and humanitarian sector SC KPI frameworks and individual metrics.

4.4 Case Study

To address our research questions, we detail a case spanning the design, development, and implementation of Supply Chain Key Performance Indicators at a large humanitarian organization.

4.4.1 Background on the Supply Chain KPI Project

The vision of the studied Supply Chain Key Performance Indicator (SC KPI) project was "to align all Supply Chain activities in order to be more effective and efficient and to transition to a culture of continuous improvement – building performance capability that can help improve service to its beneficiaries and stakeholders."

The focus of the project was on an integrated supply chain concept and on identifying metrics that would: (i) support planning, (ii) help with tactical adjustments, and (iii) inform strategic decisions. The tag-line was "SC KPIs are not statistics or evaluation reports; they are instead indicators that can be used throughout the supply chain – before, during, and after operations."

To achieve the scope and vision, the main goals of the project were:

- Define a KPI framework that can be applied across all food SC operations and services performed.
- Establish the current supply chain performance to serve as a baseline for future improvement targets.
- Provide access to KPI information – dashboards, scorecards and auto-alert reporting – to all stakeholders, especially decision makers.

- Institutionalize a continuous data-based performance improvement culture that is aligned with the organization's operational goals and strategy.

Design of KPIs for the organization's supply chain began in 2011, although a prior database migration and dashboard platform adoption phase took place starting in 2009. Throughout 2011 and 2012, a working group with members representing the disparate units managing the supply chain developed a supply chain KPI framework and KPIs for initial implementation, and the design of the dashboards began. Presently, dashboards are being developed based on the chosen KPIs. In the mid-2014, the pilot of the dashboards is scheduled to begin concurrent with the continued development of the dashboard.

While several humanitarian organizations engage in ad hoc reporting on KPIs at various operational levels, the studied SC KPI project is one of the first comprehensive and organization-wide initiatives where data is automatically queried for reporting and is delivered to users through customizable dashboards. This ongoing project is a unique collaboration between the many units of the organization that comprise its supply chain, private sector experts (current and former executives and performance measurement experts from the large multinational corporation sponsoring the project), and academic supply chain engineering experts.

4.4.2 Supply Chain KPI Implementation Phases

To address our first research question, we identify six phases of the studied SC KPI project. First, the project was initialized and organizational barriers were overcome. As with any project, this involved establishing project buy-in and ownership and defining a scope and goals. It also involved linking to the overall strategic goals of the organization, addressing the existing performance measurement mindset and culture, and overcoming inherent organizational challenges (Section 4.4.3). Second, subject to implementation constraints (Section 4.4.4.1), traditional supply chain frameworks had to be adapted to meet the needs of the humanitarian sector (Section 4.4.4.2), and in particular the organization's mandate. Third, metrics were developed within the framework and the needed data was mapped, with metric definitions being adapted as appropriate (Section 4.4.4.3).

Shifting from the initial and completed phases of the SC KPI project to the ongoing implementation, the fourth phase is the development of a dashboard, requiring extensive work from both the business and IT perspectives. The fifth phase is a pilot and dissemination of the metrics and dashboard (planned to start in mid-2014). The sixth and ongoing phase is a transition to a culture of continuous improvement by tracking the success of the SC KPIs' performance, tightening the targets, and by using the project to change the mindset.

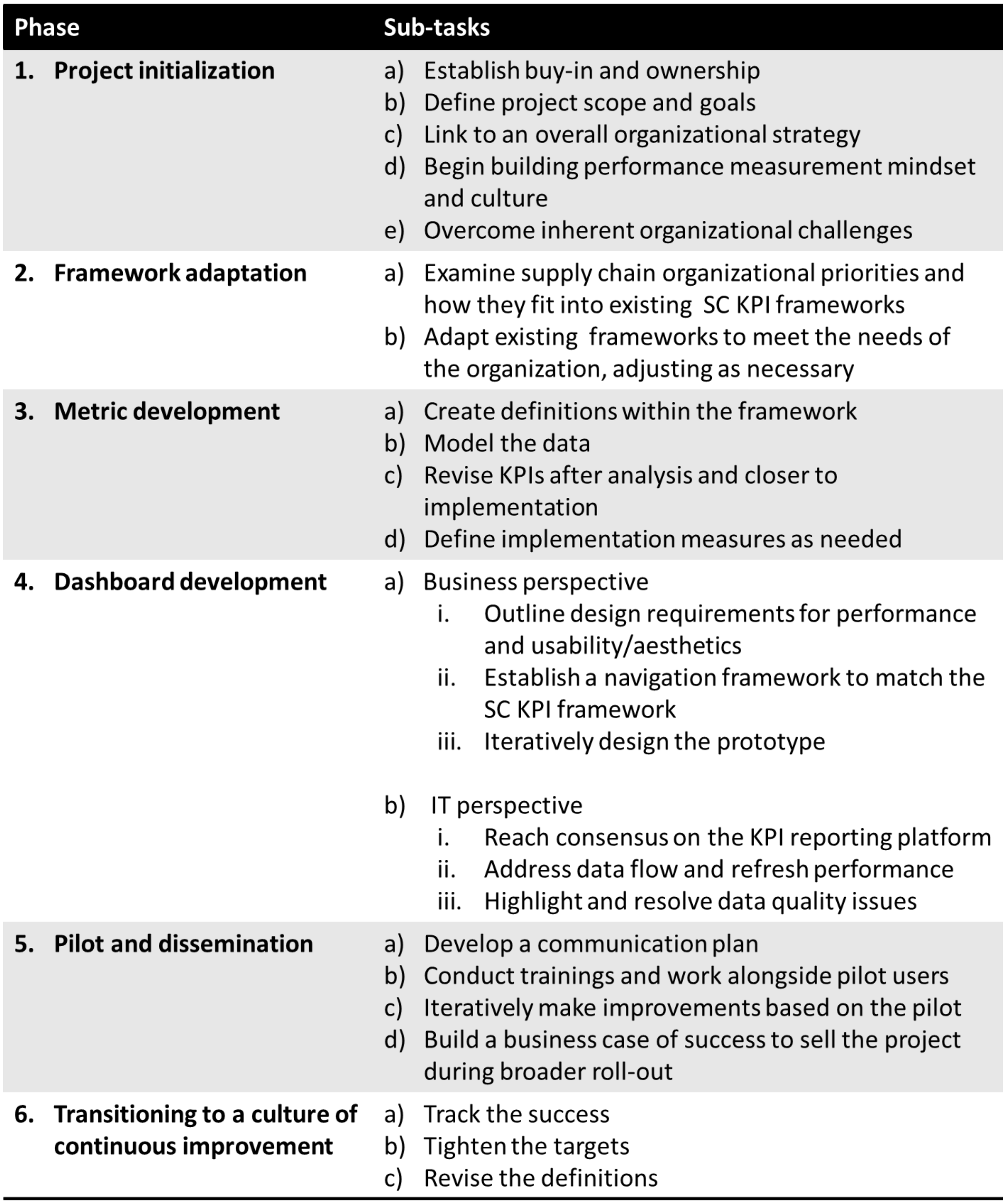
Generalizing the phases of the studied project, an abstracted process emerges for how to bring SC KPIs to an organization in a non-commercial setting. Of course, this could be another humanitarian organization, but it could also be an organization in the healthcare or public sector. The needed project phases and their main sub-tasks are summarized in Figure 27.

4.4.3 Addressing Strategy, Culture, and Organizational Structure

For our second research question, we address how the SC KPI project found its fit within corporate strategy, cultivated mindset changes, and overcame underlying organizational barriers. First, through the process of aligning with the existing and broader, strategic performance initiatives already taking place at the organization, the SC KPI project was able to find its own value-adding niche, namely, bringing together a cohesive view of supply chain operations through automated reporting. Second, a collaboration with a private sector experts helped to foster a cultural and mindset shift toward supply chain management and performance measurement. Third, we discuss how the organizational challenge of the organization's lack of a supply chain unit was addressed by the SC KPI project, and how an evolution towards more integrated supply chain management has occurred at the organization over the course of the project.

4.4.3.1 Strategic Fit: Complementing and Aligning with Broader Performance and Accountability Initiatives

At the organization, a Performance unit oversees the overall performance measurement initiatives for the organization. From the start of the SC KPI project, it was clear that any developments needed to complement and be aligned with the performance unit's work.



Phase	Sub-tasks
1. Project initialization	a) Establish buy-in and ownership b) Define project scope and goals c) Link to an overall organizational strategy d) Begin building performance measurement mindset and culture e) Overcome inherent organizational challenges
2. Framework adaptation	a) Examine supply chain organizational priorities and how they fit into existing SC KPI frameworks b) Adapt existing frameworks to meet the needs of the organization, adjusting as necessary
3. Metric development	a) Create definitions within the framework b) Model the data c) Revise KPIs after analysis and closer to implementation d) Define implementation measures as needed
4. Dashboard development	a) Business perspective i. Outline design requirements for performance and usability/aesthetics ii. Establish a navigation framework to match the SC KPI framework iii. Iteratively design the prototype b) IT perspective i. Reach consensus on the KPI reporting platform ii. Address data flow and refresh performance iii. Highlight and resolve data quality issues
5. Pilot and dissemination	a) Develop a communication plan b) Conduct trainings and work alongside pilot users c) Iteratively make improvements based on the pilot d) Build a business case of success to sell the project during broader roll-out
6. Transitioning to a culture of continuous improvement	a) Track the success b) Tighten the targets c) Revise the definitions

Figure 27: Phases of a full implementation of Supply Chain KPIs at a humanitarian organization

Since the performance unit had already created a Management Results Framework (MRF) based on the organization's strategic objectives, there was also a need to justify why the SC KPIs should be developed in addition to the KPIs already being developed within the MRF. This was a common issue that was repeatedly examined and discussed throughout initial phases of the project.

Two main reasons emerged for the value of SC KPIs in addition to the performance unit's work. First, a majority of the operational, supply chain-related KPIs in the MRF were limited in scope to the performance of individual business units (e.g. procurement or shipping). Yet, with no overall organizational supply chain unit (see Section 4.4.3.3), there was limited opportunity to capture overall supply chain performance with metrics cross-cutting the many units comprising the organization's supply chain.

Second, those KPIs within the MRF that were directly related to the supply chain were found scattered throughout the different MRF dimensions (Securing Resources, Stewardship, Learning and Innovation, Internal Business Processes, and Operational Efficiency), which further made a cohesive view of SC performance difficult. A key advantage of the SC KPI project was to bring all of the relevant metrics related to the management of the supply chain into a one-stop dashboard view.

For projects with similar objectives, there is always potential for territorial and political relationships, but instead of this, a mutually beneficial partnership developed between the performance unit's initiatives and the SC KPI project. The SC KPI project greatly benefited from (i) the pre-existing work of the performance unit to bring awareness towards KPIs and the overall value of performance measurement to the organization; (ii) the lessons shared by the performance unit about their own KPI development workshops and definition processes; and (iii) being able to be confined to a clearly defined scope of the day-to-day supply chain operations, while knowing that larger strategic performance measurement and evaluation of effectiveness was being accomplished by the performance unit. On the other hand, the performance unit benefited by having the SC KPI project examine potential new KPIs to supplement gaps in the MRF due to the lack of a supply chain unit.

4.4.3.2 Mindset and Culture: Partnering with Private-Sector Performance Measurement Experts

In general, humanitarian work is not driven by cost savings but by ensuring that beneficiaries are reached and "getting the job done" [85]. Unlike a private sector organization, the mandate of the large humanitarian organization studied is not to maximize shareholder wealth but rather to serve beneficiaries. However, donors are becoming increasingly demanding for evidence that their funding is being well-stewarded and effectively used [60, 78, 79].

Central to the success of addressing mindset and culture issues in the SC KPI project was the collaboration with private sector experts. In 2009, the organization began a focused collaboration to improve the organization's supply chain processes with the foundation of large, multinational corporation. A unique feature of this collaboration was that it involved both funding and the sharing of technical expertise from past and current executives. These executives had played a key role in a highly successful campaign to bring KPIs to their corporation's supply chain management in the 1990s and early 2000s. Their guidance in the SC KPI project played a valuable role.

Throughout the project, the past and current executives from the partnering corporation regularly offered advice and shared their expertise in how they used supply chain KPIs to transform their business. Their stories of success and their encouragement to establish a culture of improvement and accountability helped to persuade senior management to endorse the SC KPI project and helped form the scope and phases of the project. Further, their partnering expertise gave the SC KPI project a sense of credibility with stakeholders.

Communication and project presentations were important to the shaping the mindset toward the SC KPI project and helping the organization to adopt a culture of continuous improvement. Here, the partnering corporation shared advice on framing communications in terms of the 'burning platform' principle, in which a sense of urgency and need was given for adopting KPIs. They also emphasized tailoring messages to individual stakeholders according to the 'WIIFM' (What's in it for me?) principle, creating a positive sense of gain from each individual's perspective relating to their participation in the project.

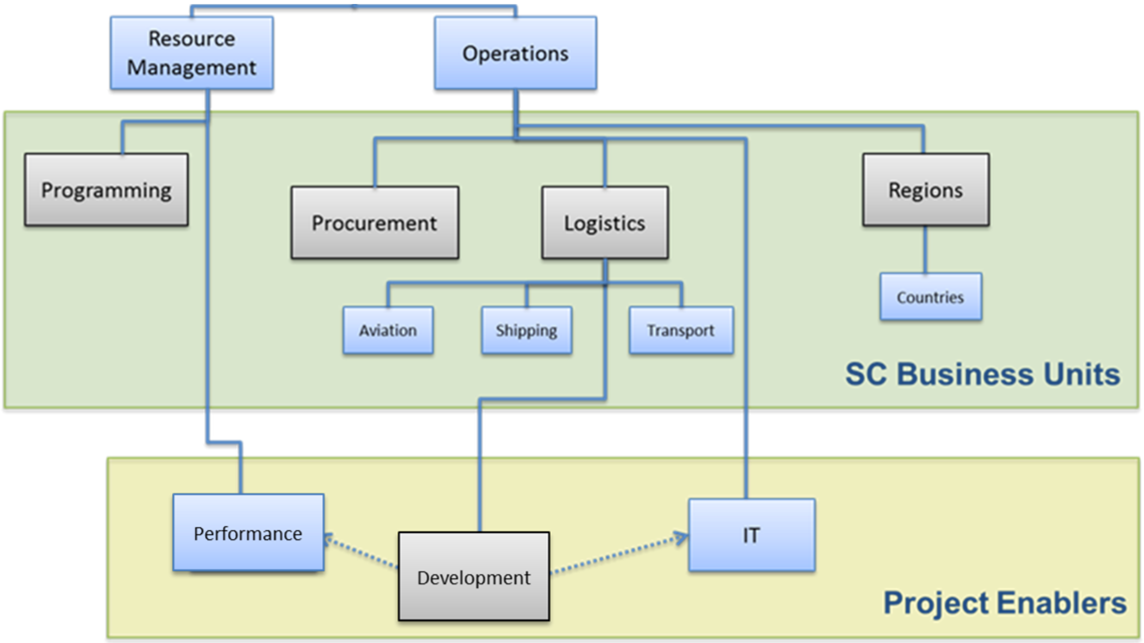


Figure 28: Disparate supply chain units and the main units facilitating the SC KPI project

4.4.3.3 Organizational Challenge: Disparate Supply Chain Units

At the large humanitarian organization studied, a challenge to supply chain management is that major decisions in the supply chain occur in separately managed units (and even divisions) of the supply chain (as illustrated in Figure 28). For example, Programming (which refers to matching donations and resources to needs and programs), Procurement, and Logistics are in different divisions, and within Logistics, Aviation, Shipping, and Overland/Inland transport are all managed separately. Additionally, there is separation between operations taking place in the field (where needs are assessed, project plans are developed, and last-mile distribution occurs) and fundraising and order-placing/management from headquarters. Due to a separation of responsibilities and information, the organization has difficulty quantifying the impact of decisions across the supply chain.

To overcome this difficulty for the SC KPI project, the first step was to form a working group consisting of focal points from all of the different supply chain business units (Programming, Procurement, Shipping, and Logistics). This working group was tasked with the objective of defining a framework and metrics to address the performance of the entire supply chain as a whole, and their work was facilitated by the SC KPI project team from the Development unit leading the implementation of the SC KPI project.

Beyond the difficulty of facilitating the creation of SC KPIs with disparate units was the longer-term challenge of who would take ownership of the SC KPI initiative after its development. Just as no single unit could create overall supply chain metrics, no single unit could adequately own the performance and management of a set of KPIs reaching beyond its scope. At the start of the SC KPI project, the resolution to this challenge was unknown, but in 2012, a key transformation happened at the organization. A formal Supply Chain Management Working Group (SCM WG) was established by the head of the organization. While not an official business unit, this group had a mandate for management decisions related to the overall supply chain.

Then, a business process review initiative in 2013-2014 resulted in the high priority recommendation to institutionalize supply chain management at the organization, and the development unit was mandated to fill a role addressing identified supply chain management initiatives. In this role, the development unit works with the SCM WG in moving forward broader institutional applications of an integrated supply chain management approach, focusing on supply chain strategy, structure, systems and tools (including the SC KPI dashboards) as well as necessary skills and mind-set. These changes have made coordination of the SC KPI project easier and established official and mandated ownership of the project.

4.4.4 Adapting Existing Supply Chain KPI Approaches

In our third research area, we examine where existing SC KPI approaches may need adaptation for the humanitarian context. First, we describe the implementation constraint of data availability and how it impacted the KPIs that were developed in the SC KPI project. Second, we examine whether and how commercial supply chain KPI frameworks can be adapted to the humanitarian context. Third, we conclude with an example of how a specific metric was adapted in order to better match the nature of the organization's business processes.

Not discussed in detail in this case, we also note that as is common in a private sector KPI implementation, the following areas of the SC KPI project took special attention to address: (i) data quality (outliers, timeliness of entry, and cleansing), (ii) high cost and lengthy time in defining business requirements and iteratively working with IT to implement the dashboard, and (iii) convincing stakeholders working in single functional areas to see the value in more holistic performance indicators for the entire supply chain. Discussion of these areas can be found in existing academic literature [21, 47, 51].

4.4.4.1 Implementation Constraint: Data Availability and the Impact on the Project Scope

Dissemination through automatically refreshing dashboards was planned in the SC KPI project from its start, requiring that displays of user-specific aggregations and filters of the KPIs (with respect to location, time, and supply chain unit) be automatically queried from corporate databases. This implementation requirement meant that all developed KPIs needed to be measurable from data in system-wide databases.

However, frequently-updated operational data at the organization is not presently available in the corporate systems for all areas of upstream planning and downstream distribution. Thus, the scope of the KPIs was limited to those that could be based on data regularly entered (e.g., within 48 hours of a transaction's occurrence) and housed in the corporate databases. For example, data tracking the distribution of specific rations to beneficiaries was unavailable to the SC KPI project, since distribution is often undertaken by cooperating partners who distribute rations on behalf of the organization and who generally have different information systems that do not link to the organization's or each others. Rather, most distribution information is tracked via a separate, less-frequent process falling under the scope of the Monitoring and Evaluation unit, for which a current initiative is seeking to improve the availability and detail of tracking.

Daily operational data availability effectively limited the scope of the supply chain covered to the areas marked with the darker blue arrow in Figure 29, from Programming through Transport, even though the organization's supply chain encompasses project planning and resource mobilization all the way through distribution to beneficiaries. However, as marked with the lighter blue arrows, an extended scope is planned where possible, or as data availability improves in the future, in order to measure as much of supply chain operations as

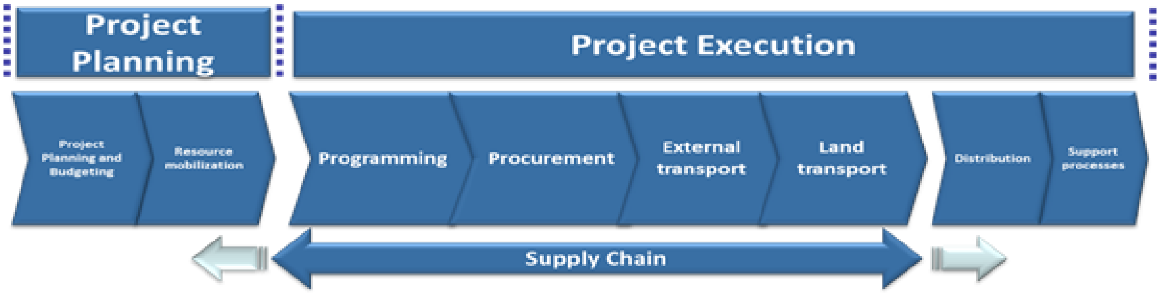
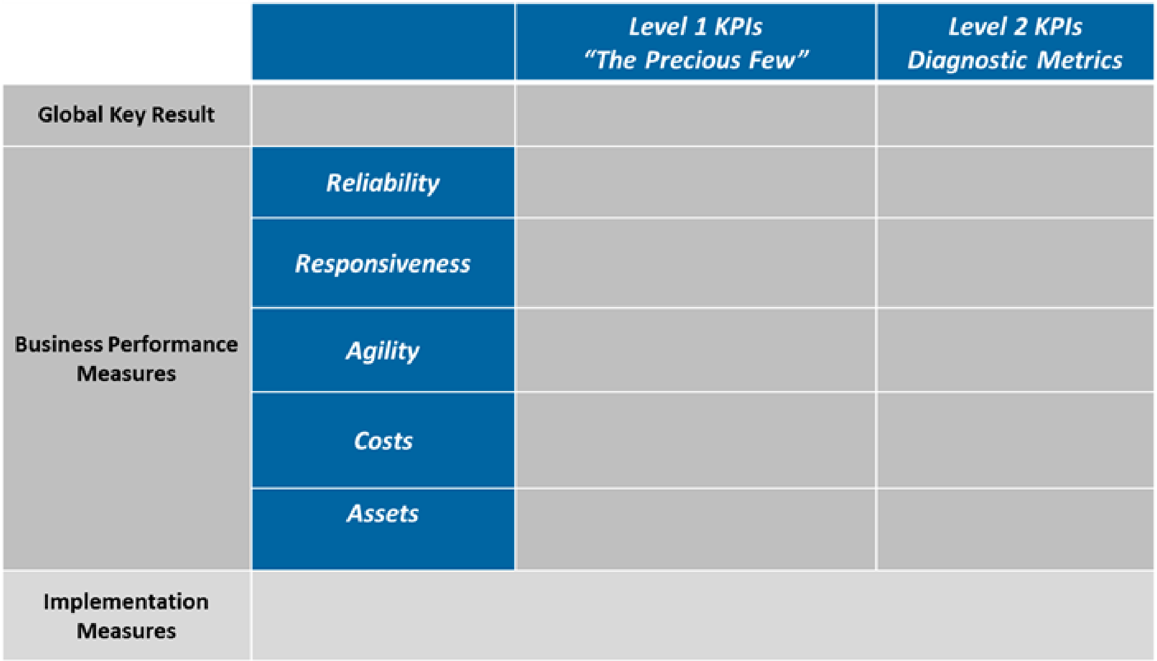


Figure 29: The organization's supply chain spans project planning and project execution, but operational data is primarily available from Programming until handover for Distribution.

possible.

Given the data availability constraints, many of the KPIs proposed in existing academic frameworks for humanitarian supply chains (e.g., [30, 70, 71]) were not seen as possible for implementation in the SC KPI project. For example, many of the proposed KPIs were based on final distribution and the impact on beneficiaries, which while critical to measuring the effectiveness of a humanitarian operation, did not match the SC KPI project's scope and the organization's data availability. Other proposed KPIs were based on critical data in the early stages of an emergency, which again is either not available or entered too late into corporate systems to make an impact on ongoing performance. Davidson [30] notes that "the inability to centrally capture time and cost data related to the procurement and distribution of goods has prevented a systematic process of performance measurement from being implemented." Thus, within the scope of the available data, KPIs were developed for the project to make the best use of the available information, which was quite extensive for the scope indicated, despite the lack of some downstream distribution data.

Here, we see a contrast with the private sector where data is increasingly available downstream through the point of sale (POS), or the point at which a customer makes a payment in exchange for goods or services [63]. Tracking POS data aids in better demand forecasting and measuring the full span of a supply chain, something not yet possible in the humanitarian sector other than through ad hoc methods of data gathering to link the entire supply chain from donor dollar to beneficiary.



		Level 1 KPIs "The Precious Few"	Level 2 KPIs Diagnostic Metrics
Global Key Result			
Business Performance Measures	Reliability		
	Responsiveness		
	Agility		
	Costs		
	Assets		
Implementation Measures			

Figure 30: Developed Humanitarian Supply Chain KPI Framework

4.4.4.2 Adapting Private Sector SC KPI Frameworks

We next examine whether and how commercial supply chain KPI frameworks can be adapted to fit the humanitarian context. The framework for the SC KPI project was adapted from two private-sector KPI systems (SCOR [28] and Global Scorecard [34]), while taking into consideration the organization's broader performance framework (the performance unit's Management Results Framework).

At the time of the project's initialization, little direction existed for how proven, private sector supply chain KPIs could or should be modified for the humanitarian context, and existing academic literature was not seen as capturing all practical needs given the organization's implementation constraints (see Section 4.4.4.1). Thus, before establishing critical metrics for the organization's supply chain, the a primary step was to create a KPI framework that could be trusted to cover the needs of the organization's supply chain. The finished result is captured in Figure 30, and we next outline how the framework was adapted.

From SCOR [28], the Supply Chain Council's performance metric framework, the structure of having supply chain attributes (Reliability, Responsiveness, Agility, Costs, and Asset

Management Effiency) to classify metrics by was used. Reliability refers to the ability to perform tasks as expected (i.e. to ensure the predictability of the outcome of a process). Responsiveness refers to the speed at which tasks are performed. Agility refers to the ability to respond to external influences. Costs refer to the cost of operating the process. Asset Management Efficiency refers to the ability to efficiently utilize assets.

Additionally, from SCOR, the division between the most critical, high-level metrics (Level 1) and further drill-down details (Level 2) is used. The private-sector experts funding the project and providing expertise especially emphasized the point of only having a small set of metrics ("the precious few" as they called them) as the drivers behind SC KPIs. These were designed to be housed in the Level 1 category.

Two key differences from the SCOR framework were established. First, the addition of a Global Key Result attribute was added. This was to ensure specific tracking of the overall mission of the organization's supply chain: (i) to have food in place according to the needs assessment and plan (which often is modified from needs assessment according to donation availability) and (ii) to ensure that this food is handed-over according to the plan either directly to beneficiaries or to cooperating partner (NGOs, governmental agencies, etc.) who then proceed with last-mile distribution to beneficiaries. Since these global key results combine traits of the majority of the traditional SCOR attributes (orders must be placed reliably, assets must be taken care of to avoid losses, responsiveness must ensure performance at the beginning of a project, etc.), it was decided to house this attribute in its own separate category.

The second difference from SCOR was that instead of having Level 2 metrics be the sub-calculations used in creating the Level 1 KPIs, it was decided that Level 2 metrics should play a more general diagnostic role. This expanded role for Level 2 Diagnostic KPIs gave the studied humanitarian organization more freedom to create the right metrics to determine root cause and the specific points in the supply chain where issues occur.

From Global Scorecard [34], the supply chain performance metric framework from the Consumer Goods Forum, the general breakdown between Business Performance Measures and Implementation Measures was used. Business Performance Measures are focused on metrics tied to supply chain performance (which includes the whole of the SCOR KPI framework and the organization's inclusion of Global Key Results). With the SC KPI project's objective of measuring supply chain business operations, it was clear that such metrics would be adopted.

What had not been considered until studying the Global Scorecard system, though, was the inclusion of Implementation Measures. Implementation Measures focus on whether the data and systems are functioning properly to effectively bring about results from the Business Performance Measures. A significant challenge was anticipated at the organization in terms of data quality and the timely completion of data entries. Thus, Implementation Measures were seen as a useful way to give specific feedback to managers about the quality of data and processes leading into the KPI data.

An extensive brainstorming exercise to map potential KPIs for the organization's supply chain to the SCOR attributes took place during the creation of the framework. The SC KPI project team classified and mapped (i) all of the KPIs from the MRF related to the supply chain, (ii) all of the Level 1 and Level 2 KPIs from SCOR deemed relevant to the organization, (iii) all of the Global Scorecard Business Performance Measures deemed relevant to the organization, and (iv) a generated set of other KPIs specific to knowledge of the organization and humanitarian operations but not yet refined by level of importance (i.e. whether they were crucial enough to be considered a Level 1, Precious KPI). The result of this process was the identification of the need for a Global Key Result attribute and the determination that other gaps in the SCOR attribute classification did not exist. The end product was also a good starting list of potential Precious KPIs, by attribute, that could later be prioritized in the process to establish the most critical Level 1 KPIs.

After the framework was created and endorsed by senior management, specific metrics were defined within each of the categories. Working with the focal points from each of the organization's supply chain business units, the development unit facilitated a draft of Level 1 KPIs. While the exact Level 1 KPIs are not yet finalized and available for publication at this time, examples include perfect order rate for Reliability (discussed next), total supply chain cost for Costs, percentage of total supply chain losses for Assets, and a measure of commodities being in-place and handed-over as planned to cooperating partners or directly distributed to beneficiaries for Global.

Then, based on the Level 1 KPIs, Level 2 KPIs were chosen to diagnose any problems in the Level 1 KPIs. For example, under perfect order rate, Level 2 KPIs diagnose whether the issue was with timing, quantity or quality, while further detailing overall supply chain lead time and underlying issues of planned order timing. For each chosen metric, an extensive data modeling exercise took place in preparation for dashboard development.

4.4.4.3 Adapting Specific Metrics: Example of the Perfect Order Rate

Perfect Order Rate was one of the first SC KPIs that was decided upon for inclusion in the set of Level 1 indicators. It is a standard measure in SCOR and is an important metric in many private sector KPI frameworks, though challenges with its implementation can arise [55]. The idea behind Perfect Order Rate made perfect sense for ensuring reliable business practices – *for all of the orders placed over a given time period, it is percentage of these orders that arrived perfectly (on-time, in the right quality, and in the right quantity, all subject to tolerances).* The more orders that arrive "perfectly," the more reliable a business is.

However, this order-based definition (where the denominator of the measure is the number of orders) did not translate well to the studied humanitarian organization's supply chain, for reasons that will next be described. In the end, the organization adapted their definition for Perfect Order Rate to a quantity-based definition (where the denominator of the measure is the tonnage ordered) that better fit the organization's business processes, *the percentage of ordered tonnage that arrived within the time tolerance in the right quality.*

When preliminary analysis was conducted on the organization's perfect order performance under the original definition, the resulting score was lower than expected and lower than what was deemed as a reasonable starting point to gain buy-in as a main measure of reliability for the organization's supply chain. The reason behind the low level of performance was the inherent difference in the types of orders placed by the studied humanitarian organization compared to those often seen at private sector organizations. The organization

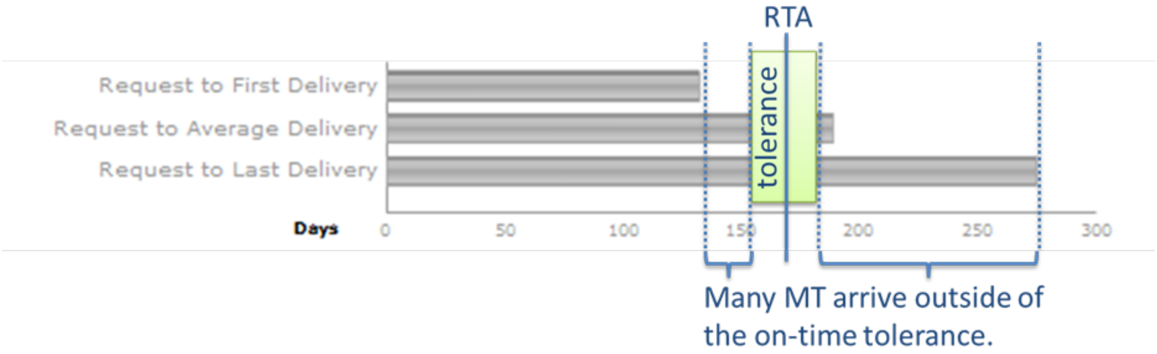


Figure 31: Due to the large orders placed by the studied humanitarian organization, deliveries take place over sometimes long time periods that extend before and after the on-time tolerance.

places large orders of bulk shipments that often arrive via many truckloads over an extended time period. Thus, a single order can have many metric tons arriving before and after the on-time tolerance (as illustrated in Figure 31). In contrast, many private sector orders arrive all at once (often in a single truckload or less), so it is more straightforward to give an exact date of arrival that can be compared to the requested date.

Two choices were available to revise Perfect Order Rate to be more useful to the organization. The first option was to maintain the standard Perfect Order Rate definition but to loosen the tolerances by widening the time window that was considered on-time and by lowering the percentage of the ordered quantity that needed to arrive in good quality within the on-time tolerance (e.g. moving from the percentage of orders with greater than 90% of the ordered quantity arriving in good quality within ± two weeks of the requested time of arrival to the percentage of orders with greater than 60% of the ordered quantity arriving in good quality within ± four weeks of the requested time of arrival).

The second option was to revise the definition to better match the nature of partial fulfillment of orders arriving over extended time periods. Instead of being centered on whether or not an entire order was considered perfect, this option shifted to explaining how much of a given order was perfect. Namely, the new option was to report the percentage of the ordered metric tonnage that arrived in good quality within the on-time tolerance. In the end, this second option was chosen because it had better performance and its resulting score gave a clearer picture of what happened.

An added strength of the revised definition is that it did not give the same weight on small orders placed at the end of a project to close out a budget as it did to the larger orders that reflect the core of the organization's supply chain operations. For a similar reason, the revised definition was less susceptible to gaming by splitting large orders into smaller ones. In theory, with a 0 or 1 score per order, a country office could improve their Perfect Order Rate score (according to the original order-based definition) simply by splitting large orders into smaller ones, e.g., by splitting an order for 10,000 metric tons into 10 orders of 1,000 metric tons. Under the original definition, the score of the order would be 0% if only 50% of the 10,000 MT arrived perfectly. On the other hand, the score could be 1 for five of split orders of 1,000 MT and 0 for the others, resulting in a score of 50% for the same resulting deliveries but based on the split orders. Under the revised definition, the score would be 50% in both the single, larger order and the smaller, split-order cases, making the adopted measure less susceptible to incentives to game the metric by changing the order size. Further study would be needed and interesting into the types of incentives relating to the other metrics chosen by the organization, especially as the SC KPI project pilots and reactionary behavior by users is observed.

4.5 Conclusion

In addressing our first research question, we identified six phases of the SC KPI project at the studied humanitarian organization and generalized them into an abstracted process for how to bring SC KPIs to a humanitarian or other non-commercial organization. These phases are given in Figure 27.

For our second research question, we addressed how the studied SC KPI project found its fit within corporate strategy, cultivated mindset changes, and overcame underlying organizational barriers. In particular, we highlighted that the support of private sector executives sharing their expertise was one of the key success factors in the early phases of the SC KPI project at organization. Such an arrangement of sharing technical expertise in addition to project funding in supply chain management is a promising direction in private sector donations, which now collectively make up the sixth largest donor group at the organization.

Third, we found that existing SC KPI approaches may need to be adapted for the humanitarian context. We explored the impact of implementation constraints and the adaptation of frameworks and metrics.

Known implementation constraints can impact the framework and metrics chosen in a performance measurement initiative. In the humanitarian sector, downstream operational and impact-related data may not be available for the last mile of distributions in a corporate database, from which standardized KPIs can be queried. This is in contrast to the private sector in which downstream data is fairly ubiquitous through increased investments in point-of-sale tracking.

The framework for the studied SC KPI project was adapted from two private-sector KPI systems (SCOR and Global Scorecard), while taking into consideration the organization's broader performance objectives. The resulting framework appears in Figure 30. We can then conclude, that for some organizations, commercial supply chain KPI frameworks be adapted to fit the humanitarian context. We also illustrated through the example of the Perfect Order Rate KPI, that metric definitions may need to be adapted to better measure how actual humanitarian supply chain operations occur.

Overall, due to extra work needed to create data foundations and develop a performance measurement mindset, supply chain KPIs may take longer to develop at a humanitarian organization than at a private-sector counterpart. However, underlying challenges like data quality and creating holistic supply chain performance goals rather than goals emphasizing unit performance are common in both contexts.

4.5.1 Limitations and Future Directions

This research is limited to the confines of a single case study, and as such further research is needed into whether and how the results may generalize. Other cases regarding implementation of performance measurement at other humanitarian or non-profit organizations would benefit the research area by providing breadth and contrast, especially undertaken in the same framework of action research. Additionally, because the SC KPI project at the studied humanitarian organization is ongoing, a follow-up study could be undertaken documenting the pilot of the dashboard, challenges in communication, training, and change management, and the resulting impact on performance and perceived usefulness of implementation.

Chapter V

CONCLUSION

In this thesis, operations research and management science techniques were applied to practical humanitarian topics, namely stable and complete assignment of staff members to field offices, bottleneck management for transportation networks, and performance measurement of the humanitarian aid supply chain. In each area, the work addressed a specific practical need: stable assignments where all agents are matched, decision support for humanitarian transport planning that includes congestion and disruptions in the system and does not assume deterministic data inputs, and implementation of performance measurement in the humanitarian supply chain. Specific results are summarized in the conclusion sections of each research area's corresponding chapter.

In this chapter, we conclude the thesis with a discussion of areas where this research may or may not generalize. While all three areas in this dissertation were motivated through observations of real operations and needs at a large humanitarian organization and were tailored to fit the humanitarian context, we have noted that the research may be applicable in other contexts as well.

The models we created for negotiated complete stable matchings in Chapter 2 can generalize to fit many existing applications of stable matching where staff members do not provide preference lists covering all jobs and thus a complete stable matching is not guaranteed. In particular, we noted that large-scale assignment of staff to jobs is important in many industries including ones such as graduates from medical school to residency or military assignment of officers, and in these large-scale instances ranking all possible agents is not realistic yet complete matchings are desired. Our studied problem was motivated by the context of the UN World Food Programme, where in order to fill the many hardship positions, negotiations and promotions are used as part of the reassignment exercises. We developed mathematical programming formulations that can be solved to optimality in reasonable time for problems the size of those faced by WFP for minimizing the number and cost of negotiations, and for larger-scale problems the size of military staff assignment, we developed algorithms that are simple to implement and solve quickly. Overall, negotiations have received little attention in the stable matching literature, and our structural results relating to the different negotiation schemes studied may contribute to interesting new research directions that could apply to many of the existing applications of stable assignments in two-sided markets.

In Chapter 3, while the congestion delay expressions, convex-cost routing model, and the simultaneous routing and investment model developed were created to address gaps in the humanitarian logistics literature, the results may also generalize to other settings. Our models capture the inherent uncertainty and disruptions that exist in many transportation networks and that may be particularly amplified in the developing world. Indeed, many types of network flow configurations are possible with the models which can be used in various public health or private sector applications. However, for the generalization of the models to hold, the underlying modeling assumptions would also need to apply to a given context, which may not always be the case. For example, a private-sector application may require a more detailed network of transport routing options, and if stations with server breakdowns are desired at multiple points on the paths between source and sink, then the provided convex-cost routing model is not appropriate without modification (e.g., specific delay expressions being developed for the given network and assumptions). We provided special focus in the chapter on a network structure relevant to humanitarian operations with congestion and off-take disruptions in the corridors concentrated in the vicinity of the ports.

Last, insights from Chapter 4 on the implementation of supply chain KPIs at a large humanitarian organization, though limited to the confines of a single case study, may be generalizable to other non-profit, public health, and even private-sector organizations. A key implementation constraint highlighted in the case was the availability of downstream data due to final distribution through cooperating partners (which often occurs at thousands of individual handover points, making complete tracking difficult and expensive). A similar constraint is faced by many large international non-governmental or public health organizations. However, at these organizations, when compared to the studied organization, a similar level of centralized data and decision-making may not be present (e.g., varying degrees of decentralization are found in different humanitarian organizations, with nearly all having field-based offices ranging in levels of autonomy and specific mandates), complicating project scope definition and organization-wide metric definition. On the other hand, at many private sector organizations, enterprise data may be more fully available and centralized, and insights from the case may generalize to private-sector organizations without access to downstream point-of-sale data (e.g., some large multinational consumer goods distributors). Overall, the level of generalization may depend on corresponding fit to the studied organization in the case in terms of the centralization of information and decision-making and the availability of downstream data.

Appendix A

PROOF APPENDIX FOR STABLE ASSIGNMENT PROBLEMS FOR STAFFING (CHAPTER 2)

A.1 Existence of Negotiated Complete Stable Matchings

A.1.1 Preliminaries

A matching μ is a set of acceptable pairs where no agent belongs to multiple pairs. For all $(i, j) \in \mu$, we define a partner function where $p_\mu(i) = j$ and $p_\mu(j) = i$.

Definition 23. Let μ and μ' be a partition of pairs (i, j) in the set $I \times J$. We define $f : (\mu, \mu') \to (x, y)$ as a function that maps matchings, $\mu \subset I \times J$, to binary decision variables, $(x, y) \in R^{|I \times J|}$. For each pair (i, j), if $(i, j) \in \mu \cap A$, then let $x_{ij} = 1$; or if $(i, j) \in \mu' \cap A$, then let $x_{ij} = 0$. Otherwise, x_{ij} is undefined (for unacceptable arcs). Let $A^c = (I \times J) \backslash A$. For each pair (i, j), if $(i, j) \in \mu \cap A^c$, then let $y_{ij} = 1$; or if $(i, j) \in \mu' \cap A^c$, then let $y_{ij} = 0$. Otherwise y_{ij} is undefined (for acceptable edges).

Lemma 24. *If μ is a stable matching then $(x, y) := f(\mu)$ satisfies:*

$$x_{ij} + \sum_{(k >_i j) \in A_i} x_{ik} + \sum_{(k >_i j) \in P_i \backslash A_i} y_{ik} + \sum_{(k >_j i) \in A_j} x_{kj} + \sum_{(k >_j i) \in P_j \backslash A_j} y_{kj} \geq 1, \forall (i, j) \in A \quad (61)$$

$$y_{ij} + \sum_{k >_i j} y_{ik} + \sum_{k >_i j} x_{ik} + \sum_{k >_j i} y_{kj} + \sum_{k >_j i} x_{kj} \geq 1, \forall (i, j) \in (I \times J) \backslash A \quad (62)$$

Proof. We first prove by contradiction that no blocking pairs exist to (x, y). In the set of unmatched pairs $\{(i, j) : x_{ij} = 0 \text{ or } y_{ij} = 0\}$, assume that there exists a blocking pair (i, j) that makes (x, y) unstable. This means that i prefers j to$i$'s partner in (x, y) and j prefers i to j's partner in (x, y). Because of the way (x, y) was constructed, $p_\mu(i) = \{k : x_{ik} = 1 \text{ or } y_{ik} = 1\}$ and $p_\mu(j) = \{k : x_{kj} = 1 \text{ or } y_{kj} = 1\}$. Thus, i prefers j to the job that i is matched to in μ, $\mu(i)$, and j prefers i to the staff member that j is matched to in μ, $\mu(j)$.

In other words, (i, j) also blocks μ. However, by assumption, μ is a stable matching with no blocking pairs, and thus we have reached our contradiction.

Next, we show through cases that no blocking pairs implies Constraints 61 and 62 hold for (x, y).

Case 1: Consider any $(i, j) \in A$ where $x_{ij} = 0$. Since we know (i, j) is not a blocking pair either one or both of the following must hold: (i) $\mu(i) >_i j$ and/or (ii) $\mu(j) >_j i$. Thus, $y_{ij} + \sum_{k>_i j} y_{ik} + \sum_{k>_i j} x_{ik} + \sum_{k>_j i} y_{kj} + \sum_{k>_j i} x_{kj} \geq 1$.

Case 2: Consider any $(i, j) \notin A$ where $y_{ij} = 0$. Since we know (i, j) is not a blocking pair either one or both of the following must hold: (i) $p_\mu(i) >_i j$ and/or (ii) $p_\mu(j) >_j i$. Thus, $x_{ij} + \sum_{(k>_i j)\in A_i} x_{ik} + \sum_{(k>_i j)\in P_i \setminus A_i} y_{ik} + \sum_{(k>_j i)\in A_j} x_{kj} + \sum_{(k>_j i)\in P_j \setminus A_j} y_{kj} \geq 1$

Case 3: For (i, j) such that $x_{ij} = 1$ or $y_{ij} = 1$, Constraints 61 and 62 trivially hold.

All cases have been covered. Thus, (x, y) satisfies Constraints 61 and 62 . □

A.1.2 Feasibility of *minNegotiations* under *Append-to-End* and *Extend-Thru* *(Proof of Theorem 1)*

Proof. (Theorem 1)

Part 1 (Extend-Thru): For a given an instance of *minNegotiations* under *Extend-Thru*, a feasible solution is constructed. Let $(x, y) := f(\mu_{complete})$, where $\mu_{complete}$ is the stable solution to the matching problem with complete preference lists, which is formed by ignoring the distinction between acceptable and unacceptable pairings for each agent and only using the complete ranked preferences. $\mu_{complete}$ is guaranteed to exist by (Gale Shapely 1962). We note that (x, y) satisfies Constraints 7 and 8, since $\mu_{complete}$ is a matching of cardinality $N = |I| = |J|$ pairs of matches. Also, the binary constraints, Constraints 13 and 14, are satisfied by construction of (x, y).

Invoking Lemma 24, Constraints13 and 14 are satisfied. Thus, all constraints in *minNegotiations* under *Extend-Thru* are satisfied, and (x, y) is a feasible solution to *minNegotiations* under *Extend-Thru* .

Part 2 (Append-to-End): Since constraints of *minNegotiations* under *Append-to-End* are a subset of the constraints of *minNegotiations* under *Extend-Thru*, it immediately follows from the above that *minNegotiations* under *Append-to-End* is feasible. □

A.2 Minimizing the Number of Negotiations under Append-to-End (*relevant results for the proof of Theorem 4*)

A.2.1 Classic Theorem and New Corollaries

Theorem 25. *(from Gusfield and Irving [38]: Theorem 1.4.3) If, in a stable matching instance, some staff i appends a previously unacceptable job j to the end of his list, then in both the staff and job-optimal stable matchings for the extended instance, no job is worse off and no staff member, except possibly staff i, is better off.*

The following two simple corollaries do not appear in [38]. It is unknown whether they appear in other works. These corollaries are useful in bounding the increase in cardinality from appending preferences.

Corollary 26. *In a stable matching instance, if staff i appends a previously unacceptable agent to the end of his list, then the cardinality of the number of pairs matched in the extended instance increases by at most one pair compared to the original instance.*

Proof. Assume, for the sake of contradiction, that the cardinality of matched pairs increases by 2 or more in the extended instance. Then, at least 2 staff went from being unmatched to matched. By definition, we say that being matched is preferred to being unmatched. Thus, at least 2 staff became better off in the extended instance, which contradicts Theorem 25. □

Corollary 27. *In a stable matching instance, if staff i appends a previously unacceptable agent to the end of his list, then i is the only staff who can be unmatched in the original instance and matched in the extended instance.*

A.2.2 Bounding the Impact of Negotiations

First, we give a tighter bound on the increase in cardinality compared to Corollary 26 for the case when the agent that appends its list is already matched in the original stable matching instance.

Lemma 28. *In a stable matching instance, if some staff i appends a previously unacceptable job j to the end of his list, and if i is already matched in the original instance, then the* *cardinality of the number of pairs matched in the extended instance is equivalent to the original instance*

Proof. Since staff i has a stable partner in the original instance, then the staff-oriented version of the algorithm for the extended instance will be identical to that for the original instance, terminating before i reaches j in his list. Thus the staff-optimal stable matching is unchanged and has the same cardinality in the extended instance as in the original. □

Recall that P_i is agent i's ordered preference list, and that A_i is a truncated version of P_i such that $k \in A_i$ if and only if $i \in P_k$ and $k \in P_i$ (i.e. $(i, k) \in A$, if i is a staff member), with the ordering in P_i carried over to A_i.

Theorem 29. *Consider a stable matching instance where M is the cardinality of the pairs matched. To create an extended instance, pick some $(i, j) \notin A$. If $j \notin P_i$, then append j to the end of P_i. Similarly, if $i \notin P_j$, then append i to the end of P_j. This extended stable match instance will have at most cardinality M+1 pairs matched.*

Proof. Case 1: single append ($j \notin P_i$ OR $i \notin P_j$)

By Corollary 26, the extended stable match instance will have at most cardinality M+1 pairs matched.

Case 2: double append ($j \notin P_i$ AND $i \notin P_j$)

Let P refer to the original stable matching preference. Let P'' refer to the extended instance. Create P' by starting with P and appending j to the end of P_i (and note that P' differs from P'' by not additionally appending i to P_j).

First, we show that stable matching solutions to P' have cardinality M pairs. If i is matched in the stable matching solutions to P, then the cardinality of the stable matching solutions to P' is M by Lemma 28. On the other hand, if i is unmatched in the stable matching solutions to P, then the possibility that i becomes matched in the stable matchings of P' needs to be investigated, since by Corollary 27 staff i is the only possible staff member that could go from unmatched to matched in P'.

From (Thm. 1.2.2, Gusfield-Irving 1989), we know that all possible executions of the

Gale-Shapely algorithm (with the staff members as proposers) yield the same stable matching. Thus, the staff member proposing in any iteration of the algorithm can be any staff member who is still unassigned and has not exhausted his/her preference list. For the extended instance, assume that the staff-proposing algorithm proceeds the same as in the original instance, leaving i as the only unmatched staff member with a single proposal left to try, namely to j. When the algorithm proceeds with i's proposal to j, j will reject i due to unacceptability (since $j \notin P_i$), and the algorithm will conclude, with an unchanged staff-optimal stable matching in P and P' of cardinality M pairs.

Thus, whether i is unmatched or matched in the stable matching solution to P, the stable matching solution to P' has cardinality M pairs. To conclude the proof, we note that P'' is an extended instance of P' where job j appends previously unacceptable staff member i to the end of P'_j. From Corollary 26, we conclude that the stable matching solutions to P'' has at most cardinality $M + 1$ pairs. □

Corollary 30. *Consider a stable matching instance where M is the cardinality of the pairs matched. To create an extended instance, pick some K pairs in* $(I \times J) \backslash A$, such that at most one edge is incident to each agent. *For each pair,* (i, j), *chosen, if $j \notin P_i$, then append j to the end of P_i. Similarly, if $i \notin P_j$, then append i to the end of P_j. This extended stable match instance will have at most cardinality M+K pairs matched.*

Proof. Theorem 29 can be applied sequentially K times for each of the K arcs, producing a bound of $M + K$ on the extended instance. Since at most one edge is incident to each agent, the order that the pairs are chosen and the lists appended does not matter. The final preference lists in the extended instance will always be the same. □

A.2.3 Lower Bound of $N - M$ to $minNegotiations$ under ***Append-to-End***

We next build on Theorem 29 (and Corollary 30) which bounds the increase in cardinality per set of preference list appendages accompanying unacceptable arcs.

Lemma 31. *For a stable matching instance, if M is the cardinality of the stable matching(s) in the instance, then $N - M$ is a lower bound on the optimal solution value in $minNegotiations$ under Append-to-End .*

Proof. Let (x, y) be a feasible, integer solution to *minNegotiations* under *Append-to-End.* By Constraints 7 and 8, at most one edge is incident to each agent and all agents are matched in N pairs. Let $K = \sum_{(i,j)\notin A} y_{ij}$, be the number of unacceptable pairs chosen in the solution. Assume for the sake of contradiction that $K \leq N - M - 1$. However, by Corollary 30, if an extended instance is created based on the K chosen unacceptable pairs (according to the method in the corollary), then at most $M + K \leq N - 1$ pairs can be stable in the extended instance. This contradicts that (x, y) is a feasible solution to *minNegotiations* under *Append-to-End* that matches N pairs. □

A.3 Integral Polyhedron for the Linear Relaxation of **minNegotiations** *under* **Extend-Thru** *(Proof of Theorem* **6)**

Proof. (Theorem *6*)

Given an instance of *minNegotiations* (under the negotiation scheme Extend-Thru) where A denotes the set of acceptable pairs in the instance, let $Q \subseteq R^{|I\times J|}$ be defined as the feasible region of the linear relaxation of the model.

$$(Q) \qquad \textstyle\sum_{j\in J\setminus A_i} y_{ij} + \sum_{j\in A_i} x_{ij} = 1 \qquad \forall i \in I \tag{63}$$

$$\textstyle\sum_{i\in I\setminus A_j} y_{ij} + \sum_{i\in A_j} x_{ij} = 1 \qquad \forall j \in J \tag{64}$$

$$x_{ij} + \textstyle\sum_{k>_i j} x_{ik} + \sum_{k>_j i} x_{kj} \geq 1 \qquad \forall (i,j) \in A \tag{65}$$

$$y_{ij} + \textstyle\sum_{k>_i j} y_{ik} + \sum_{k>_i j} x_{ik} + \sum_{k>_j i} y_{kj} + \sum_{k>_j i} x_{kj} \geq 1 \quad \forall (i,j) \in (I \times J)\setminus A \tag{66}$$

$$0 \leq x_{ij} \leq 1 \qquad \forall (i,j) \in A \tag{67}$$

$$0 \leq y_{ij} \leq 1 \qquad \forall (i,j) \in (I \times J)\setminus A \tag{68}$$

For the same instance, let $P \subseteq R^{|I\times J|}$ be defined as the feasible region to the complete, stable matching problem in Formulation 1 (Constraints 2-5) in which complete preferences lists are used for each agent, constructed as the ranked ordering of acceptable pairs followed by the ranked ordering of unacceptable pairs for each agent.

$$(P) \qquad \textstyle\sum_{j\in J} z_{ij} \leq 1 \qquad \forall i \in I \tag{69}$$

$$\textstyle\sum_{i\in I} z_{ij} \leq 1 \qquad \forall j \in J \tag{70}$$

$$z_{ij} + \textstyle\sum_{k>_i j} z_{ik} + \sum_{k>_j i} z_{kj} \geq 1 \quad \forall (i,j) \in I \times J \tag{71}$$

$$0 \leq z_{ij} \leq 1 \qquad \forall (i,j) \in I \times J \tag{72}$$

Next, we show that Q is a face of P by applying a change of variables to P (to match the variables used in Q) and invoking the definition of a face. Let $x_{ij} = z_{ij}, \forall (i,j) \in A$, and let $y_{ij} = z_{ij}, \forall (i,j) \in (I \times J)\backslash A$. For $(i,j) \in (I \times J)\backslash A$, x_{ij} does not exist, and for $(i,j) \in A$, y_{ij} does not exist, so that $z \in R^{|I|^2}$ and $(x,y) \in R^{|I|^2}$. Then, P can be rewritten as follows:

$$(P) \qquad \textstyle\sum_{j\in J\backslash A_i} y_{ij} + \sum_{j\in A_i} x_{ij} \leq 1 \qquad \forall i \in I \tag{73}$$

$$\textstyle\sum_{i\in I\backslash A_j} y_{ij} + \sum_{i\in A_j} x_{ij} \leq 1 \qquad \forall j \in J \tag{74}$$

$$x_{ij} + \textstyle\sum_{k>_i j} x_{ik} + \sum_{k>_j i} x_{kj} \geq 1 \qquad \forall (i,j) \in A \tag{75}$$

$$y_{ij} + \textstyle\sum_{k>_i j} y_{ik} + \sum_{k>_i j} x_{ik} + \sum_{k>_j i} y_{kj} + \sum_{k>_j i} x_{kj} \geq 1 \quad \forall (i,j) \in (I \times J)\backslash A \tag{76}$$

$$0 \leq x_{ij} \leq 1 \qquad \forall (i,j) \in A \tag{77}$$

$$0 \leq y_{ij} \leq 1 \qquad \forall (i,j) \in (I \times J)\backslash A \tag{78}$$

The equivalencies Constraints 73 and 74 with Constraints 69 and 70, respectively, can easily be seen as can the equivalency between Constraints 77 and 78 with Constraint 72. The equivalency between Constraints 75 and 76 and Constraint 76 can be seen through the following two cases which cover the full constraint set expressed in Constraint 71.

Case 1: $(i,j) \in A$

$z_{ij} + \sum_{k>_i j} z_{ik} + \sum_{k>_j i} z_{kj} \geq 1 \Longleftrightarrow x_{ij} + \sum_{k>_i j} x_{ik} + \sum_{k>_j i} x_{kj} \geq 1$

The y variables corresponding to unacceptable arcs do not appear since $(i,j) \in A$ implies that if i prefers k to j then $(i,k) \in A$. Likewise, $(i,j) \in A$ implies that if j prefers k to i then $(k,j) \in A$.

Case 2: $(i,j) \in (I \times J)\backslash A$

$$
\begin{aligned}
& z_{ij} + \textstyle\sum_{k>_i j} z_{ik} + \sum_{k>_j i} z_{kj} && \geq 1 \\
\iff \quad & y_{ij} + \left(\textstyle\sum_{k>_i j} y_{ik} + \sum_{k>_i j} x_{ik}\right) + \left(\sum_{k>_j i} y_{kj} + \sum_{k>_j i} x_{kj}\right) && \geq 1 \\
\iff \quad & y_{ij} + \left(\textstyle\sum_{k>_i j} y_{ik} + \sum_{k>_i j} x_{ik}\right) + \left(\sum_{k>_j i} y_{kj} + \sum_{k>_j i} x_{kj}\right) && \geq 1
\end{aligned}
$$

By (Shrijver 2000), for some polyhedron $S = \{x | Ax \leq b\}$, F is a face of S if and only if F is non-empty and $F = \{x \in S | A'x = b'\}$ for some subsystem $A'x \leq b'$ of $Ax \leq b$. With P as expressed by Constraints 73 - 78, it can be seen that Q is a face of P where feasible solutions to Q satisfy all constraints in P, Constraints 73 and 74 form the required subsystem, and Q is know to be non-empty from Theorem 1.

Concluding the proof, since Q is a face of P, we obtain that the set of vertices of Q is contained in the set of vertices of P. Thus, since P is known to be an integral polyhedron (Roth et al. 1993), Q must also be an integral polyhedron. □

A.4 Minimizing the Number of Negotiations under Move-to-Beginning (*relevant results for the proof of Theorem 12*)

A.4.1 Bounding the Impact of Negotiations

Theorem 32. *Consider a stable matching instance where M is the cardinality of the pairs matched. To create an extended instance, pick some $(i, j) \notin A$. Append j to the front of P_i, and append i to the front of P_j. This extended stable match instance will have at most cardinality M+1 pairs matched.*

Proof. From (Thm. 1.2.2, [38]), we know that all possible executions of the Gale-Shapely algorithm (with the staff members as proposers) yield the same stable matching. Thus, the staff member proposing in any iteration of the algorithm can be any staff member who is still unassigned and has not exhausted his/her preference list. For the extended instance, assume the staff-proposing algorithm runs until all but staff member i's full list of proposals remains. This means that every other staff member is either matched or out of proposals. At this point in the algorithm, let K be the number of pairs matched.

We note that $K \leq M$, otherwise the original stable matching solution would have to match more than M pairs, since i's ensuing proposals (with both the original and the extended preference list) could not lead to a decrease in the cardinality of pairs matched. The addition of i to the front of j's preference won't have impacted the algorithm at this stage, since no proposals from i have occurred yet, and the rest of j's preferences after i in the extended instance exactly match j's original preference list.

Then, the algorithm proceeds with i's proposal to j, the top-ranked agent on i's list in the extended instance. Job j will accept, since i is j's top choice. If j was not engaged at the time of the proposal, the there are no other staff members eligible to propose and algorithm concludes at most $M + 1$ pairs matched. On the other hand, if j was engaged at the time of i's proposal and rejects this lower ranked agent k by accepting staff member i, then k can proceed with with proposing to the next job on k's preference list. From here, a cascading effect of broken engagements and proposals can occur to conclude the algorithm. However, after each broken proposal, K pairs are matched and at most one staff member has any proposals remaining, thus the number of pairs matched cannot exceed $K + 1 \leq M + 1$. □

Corollary 33. *Consider a stable matching instance where M is the cardinality of the pairs matched. To create an extended instance, pick some K pairs in* $(I \times J)\backslash A$, such that at most one edge is incident to each agent. *For each pair,* (i, j), *chosen, append j to the front of* P_i *and append i to the front of* P_j. *This extended stable match instance will have at most cardinality M+K pairs matched.*

Proof. Theorem 32 can be applied sequentially K times for each of the K arcs, producing a bound of $M + K$ on the extended instance. Since at most one edge is incident to each agent, the order that the pairs are chosen and the lists appended does not matter. The final preference lists in the extended instance will always be the same. □

A.4.2 Lower Bound of $N - M$ to $minNegotiations$ under *Move-to-Beginning*

Lemma 34. *For a stable matching instance, if M is the cardinality of the stable matching(s) in the instance, then N − M is a lower bound on the optimal solution value in* $minNegotiations$ *under Move-to-Beginning.*

Proof. Let (x, y) be a feasible, integer solution to $minNegotiations$ under *Move-to-Beginning*. By Constraints 19 and 20, at most one edge is incident to each agent and all agents are matched in N pairs. Let $K = \sum_{(i,j)\in A} z_{ij} + \sum_{(i,j)\notin A} y_{ij}$, be the number pairs with negotiated top preferences in the solution. Assume for the sake of contradiction that $K \leq N - M - 1$. However, by Corollary 33, if an extended instance is created based on the K chosen unacceptable pairs (according to the method in the corollary), then at most $M + K \leq N - 1$ pairs can be stable in the extended instance. This contradicts that (x, y) is a feasible solution to Model 4 that matches N pairs. □

Appendix B

CONVEX COST ROUTING MODEL WITH INLAND DELIVERIES

A Discharge Port Routing and and Inland Delivery Network appears in Figure 32, where N discharge ports are available to route a total of λ flow (e.g,. the monthly demand) through and onward to M delivery points on delivery arcs, $D = \{1, ..., N\} \times \{1, ..., M\}$. Each port-corridor $i \in \{1, ..., N\}$ is characterized by its service and failure rate parameters, μ_{pi}, μ_{ci}, f_i, r_i, and v_i, and each delivery point $j \in \{1, ..., M\}$ has demand $b(j)$. We assume that the supply is equal to the demand ($\lambda = \sum_{j \in \{1,...,M\}} b(j)$). A mathematical program with objective (79), with convex costs on each arc, can be solved to find the minimum cost flow through the network, where scalars α_i, β_{pi} and β_{ci}, and γ_{ij} quantify port fees, port and corridor delay costs, and delivery costs, respectively.

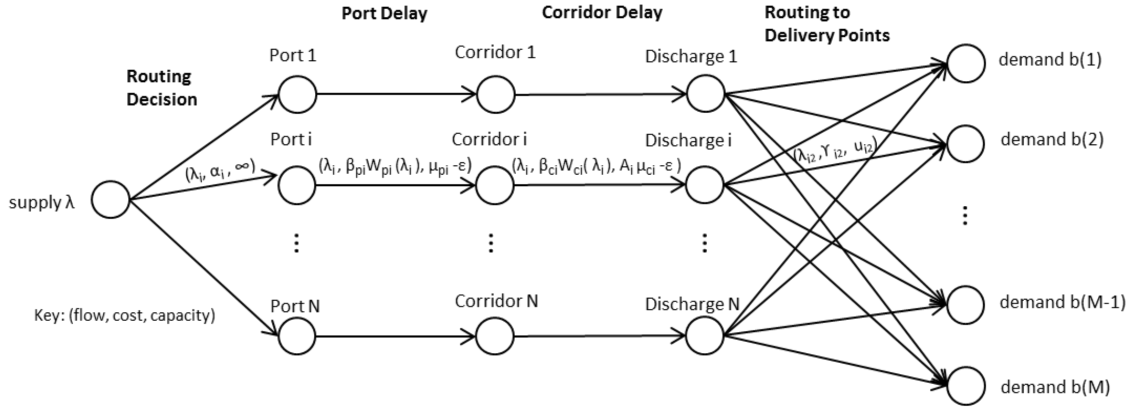


Figure 32: Discharge Port Routing and and Inland Delivery Network

Formulation 8 Convex Cost Routing Model with Inland Deliveries (corresponding to Figure 32)

$$
\begin{aligned}
min \quad & \textstyle\sum_{i\in\{1,...,N\}} \lambda_i(\alpha_i + \beta_{pi}W_{pi} + \beta_{ci}W_{ci}) + \sum_{(i,j)\in D} \gamma_{ij}\lambda_{ij} & & (79)\\
s.t. \quad & \textstyle\sum_{i\in\{1,...,N\}} \lambda_i = \lambda & & (80)\\
& \textstyle\sum_{j\in\{1,...,M\}} \lambda_{ij} - \lambda_i = 0 & \forall i \in \{1,...,N\} & \quad (81)\\
& \textstyle\sum_{i\in\{1,...,N\}} \lambda_{ij} = -b(j) & \forall j \in \{1,...,M\} & \quad (82)\\
& 0 \le \lambda_i \le \mu_{ei} - \epsilon & \forall i \in \{1,...,N\} & \quad (83)\\
& 0 \le \lambda_{ij} \le u_{ij} & \forall (i,j) \in D & \quad (84)
\end{aligned}
$$

Note that due to the network structure, a single flow variable, λ_i can represent flow from the routing node through the port and corridor ($\lambda_i =: \ \lambda_{0,pi} = \lambda_{pi,ci} = \lambda_{ci,di}$). The model also includes constraints to ensure (i) flow balance for routing (80), discharge (81) and delivery (82) nodes and (ii) capacity on the arcs is not exceeded for each port-corridor (83), where $\mu_{ei} = min(\mu_{pi},\, A_i\mu_{ci})$ is the path effective processing rate, and (iii) capacity on the delivery arcs is not exceeded (84). For a feasible solution to exist, the overall network capacity must be sufficient to handle the flow in order, $\sum_i \mu_{ei} > \lambda$ and $\sum_i u_{ij} \ge b(j),\ \forall j \in 1, ..., N$.

REFERENCES

[1] Abdulkadiroglu, A., Pathak, P. A., and Roth, A. E., "Strategy-proofness versus Efficiency in Matching with Indifferences: Redesigning the NYC High School Match," *American Economic Review*, vol. 99, pp. 1954–1978, Dec. 2009.

[2] Abdulkadiroglu, A. and Sonmez, T., "School choice: A mechanism design approach," *The American Economic Review*, vol. 93, no. 3, pp. 729–747, 2003.

[3] Abidi, H. and Klumpp, M., "Performance measurement in humanitarian logistics: A literature," Working paper.

[4] Afshar, A. and Haghani, A., "Modeling integrated supply chain logistics in real-time large-scale disaster relief operations," *Socio-Economic Planning Sciences*, vol. 46, no. 4, pp. 327–338, 2012.

[5] Ahmed, S., King, A. J., and Parija, G., "A multi-stage stochastic integer programming approach for capacity expansion under uncertainty," *Journal of Global Optimization*, vol. 26, no. 1, pp. 3–24, 2003.

[6] Ahuja, R. K., Magnanti, T. L., and Orlin, J. B., "Network flows: theory, algorithms, and applications," 1993.

[7] Altendorfer, K. and Minner, S., "Simultaneous optimization of capacity and planned lead time in a two-stage production system with different customer due dates," *European Journal of Operational Research*, vol. 213, no. 1, pp. 134–146, 2011.

[8] Alvarenga, R., Ergun, O., Li, J., Mata, F., Shekhani, N., Slaton, D., Stone, J., Vasudevan, A., and Yang, E., "World food programme east african corridor optimization," tech. rep., Senior Design Final Report, H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA, 2010.

[9] Arvis, J.-F., Raballand, G., and Marteau, J.-F., "The cost of being landlocked: logistics costs and supply chain reliability," *World Bank Policy Research Working Paper*, no. 4258, 2007.

[10] Arzu Akyuz, G. and Erman Erkan, T., "Supply chain performance measurement: a literature review," *International Journal of Production Research*, vol. 48, no. 17, pp. 5137–5155, 2010.

[11] BBC, "What is delaying haiti's aid?," January 21 2010.

[12] BBC, "Syria conflict: Homs aid convoy comes under fire," February 8 2014.

[13] Beamon, B. M. and Balcik, B., "Performance measurement in humanitarian relief chains," *International Journal of Public Sector Management*, vol. 21, no. 1, pp. 4–25, 2008.

[14] Berkoune, D., Renaud, J., Rekik, M., and Ruiz, A., "Transportation in disaster response operations," *Socio-Economic Planning Sciences*, vol. 46, no. 1, pp. 23–32, 2012.

[15] Bertsimas, D. and Tsitsiklis, J. N., "Introduction to linear optimization," 1997.

[16] Bhat, U. N., *An introduction to queueing theory: modeling and analysis in applications.* Springer, 2008.

[17] Biró, P., Manlove, D. F., and Mittal, S., "Size versus stability in the marriage problem," *Theoretical Computer Science*, vol. 411, pp. 1828–1841, Mar. 2010.

[18] Bish, D. R., Chamberlayne, E. P., and Rakha, H. A., "Optimizing network flows with congestion-based flow reductions," *Networks and Spatial Economics*, vol. 13, no. 3, pp. 283–306, 2013.

[19] Bourne, M., Mills, J., Wilcox, M., Neely, A., and Platts, K., "Designing, implementing and updating performance measurement systems," *International Journal of Operations & Production Management*, vol. 20, no. 7, pp. 754–771, 2000.

[20] Boyd, S. P. and Vandenberghe, L., *Convex optimization.* Cambridge university press, 2004.

[21] Braz, R. G. F., Scavarda, L. F., and Martins, R. A., "Reviewing and improving performance measurement systems: An action research," *International Journal of Production Economics*, vol. 133, no. 2, pp. 751–760, 2011.

[22] Carey, M., "Optimal time-varying flows on congested networks," *Operations research*, vol. 35, no. 1, pp. 58–69, 1987.

[23] Chiang, M., Sutivong, A., and Boyd, S., "Efficient nonlinear optimizations of queuing systems," in *Global Telecommunications Conference, 2002. GLOBECOM'02. IEEE*, vol. 3, pp. 2425–2429, IEEE, 2002.

[24] Choi, A. K., Beresford, A. K., Pettit, S. J., and Bayusuf, F., "Humanitarian aid distribution in east africa: A study in supply chain volatility and fragility," in *Supply Chain Forum: An International Journal*, vol. 11, pp. 20–31, KEDGE Business School, 2010.

[25] Cluster, L., "Logistics capacity assessment - djibouti port of djibouti s.a (paid)," 2013.

[26] Coles, P. and Shorrer, R., "Optimal truncation in matching markets," 2013.

[27] Coughlan, P. and Coghlan, D., "Action research for operations management," *International journal of operations & production management*, vol. 22, no. 2, pp. 220–240, 2002.

[28] Council, S. C., "Supply chain operations reference model: Version 10.0," *URL: https://supply-chain.org/scor/*, 2010.

[29] Daganzo, C. F., "The cell transmission model, part ii: network traffic," *Transportation Research Part B: Methodological*, vol. 29, no. 2, pp. 79–93, 1995.

[30] Davidson, A. L., *Key performance indicators in humanitarian logistics.* PhD thesis, Massachusetts Institute of Technology, 2006.

[31] De Angelis, V., Mecoli, M., Nikoi, C., and Storchi, G., "Multiperiod integrated routing and scheduling of world food programme cargo planes in angola," *Computers & Operations Research*, vol. 34, no. 6, pp. 1601–1615, 2007.

[32] Ehlers, L., "Truncation strategies in matching markets," *Mathematics of Operations Research*, vol. 33, no. 2, pp. 327–335, 2008.

[33] Ergun, O., Keskinocak, P., and Swann, J., "Logistics ignored in disaster relief," June 11 2010.

[34] Forum, C. G., "Global scorecard: Kpi definitions 2.10," *URL: http://www.globalscorecard.net/live/*, 2009.

[35] Gale, D. and Shapley, L., "College admissions and the stability of marriage," *The American Mathematical Monthly*, vol. 69, no. 1, pp. 9–15, 1962.

[36] Gale, D. and Sotomayor, M., "Ms. machiavelli and the stable matching problem," *American Mathematical Monthly*, vol. 92, no. 4, pp. 261–268, 1985.

[37] Gunasekaran, A. and Kobu, B., "Performance measures and metrics in logistics and supply chain management: a review of recent literature (1995–2004) for research and applications," *International Journal of Production Research*, vol. 45, no. 12, pp. 2819–2840, 2007.

[38] Gusfield, D. and Irving, R. W., *The stable marriage problem: structure and algorithms*, vol. 54. MIT press Cambridge, 1989.

[39] Hamada, K., Iwama, K., and Miyazaki, S., "An improved approximation lower bound for finding almost stable maximum matchings," *Information Processing Letters*, vol. 109, pp. 1036–1040, Aug. 2009.

[40] Hopp, W. J. and Spearman, M. L., *Factory physics.* Waveland Press, 2011.

[41] Institute, F., "Logistics and the effective delivery of humanitarian relief," 2005.

[42] Jeong, K.-Y., Hong, J.-D., and Xie, Y., "Design of emergency logistics networks, taking efficiency, risk and robustness into consideration," *International Journal of Logistics Research and Applications*, vol. 17, no. 1, pp. 1–22, 2014.

[43] Kaplan, R. S. and Norton, D., "The balanced scorecard: Measures that drive performance," *Harvard Business Review*, vol. 70, no. 1, pp. 71–79, 1992.

[44] Komrska, J., Kopczak, L. R., and Swaminathan, J. M., "When supply chains save lives," *Supply Chain Management Review*, vol. 17, no. 1, pp. 42–49, 2013.

[45] Lewis, B. M., Erera, A. L., Nowak, M. A., and Chelsea C III, W., "Managing inventory in global supply chains facing port-of-entry disruption risks," *Transportation Science*, vol. 47, no. 2, pp. 162–180, 2013.

[46] Liu, T. and Lam, J. S. L., "Impact of port disruption on transportation network,"

[47] Lohman, C., Fortuin, L., and Wouters, M., "Designing a performance measurement system: a case study," *European Journal of Operational Research*, vol. 156, no. 2, pp. 267–286, 2004.

[48] Manlove, D., *Algorithmics of matching under preferences.* World Scientific Publishing, 2013.

[49] Maspero, E. L. and Ittmann, H. W., "Rise of humanitarian logistics," 2008.

[50] Mohring, F. and Link, D., "Get seaports ready for disaster — strengthening preparedness at african seaports by improving performance," in *Managing Humanitarian Supply Chains — Strategies, Practices and Research* (Hellingrath, B., Link, D., and Widera, A., eds.), pp. 33–45, 2013. Publication status: Published.

[51] Neely, A., "The performance measurement revolution: why now and what next?," *International Journal of Operations & Production Management*, vol. 19, no. 2, pp. 205–228, 1999.

[52] Neely, A., “The evolution of performance measurement research: developments in the last decade and a research agenda for the next,” *International Journal of Operations & Production Management*, vol. 25, no. 12, pp. 1264–1277, 2005.

[53] Nemhauser, G. L. and Wolsey, L. A., *Integer and combinatorial optimization*, vol. 18. New York: Wiley, 1988.

[54] NEOS, “Neos server for baron,” 2014.

[55] Novack, R. A. and Thomas, D. J., “The challenges of implementing the perfect order concept,” *Transportation Journal*, pp. 5–16, 2004.

[56] of Science, T. R. S. A., “Scientific Background on the Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel,” vol. 50005, 2012.

[57] of Science, T. R. S. A., “Stable matching: Theory, evidence, and practical design,” 2012.

[58] of the United States Navy, O. W., “Status of the Navy: Navy Personnel,” 2014.

[59] of World Food Programme, F. C., “Report on the implementation of the external auditor recommendations,” 2013.

[60] Oloruntoba, R. and Gray, R., “Customer service in emergency relief chains,” *International Journal of Physical Distribution & Logistics Management*, vol. 39, no. 6, pp. 486–505, 2009.

[61] Optimization, G., “Gurobi optimizer 5.6 reference manual,” *URL: http://www.gurobi.com*, 2013.

[62] Ordóñez, F. and Zhao, J., “Robust capacity expansion of network flows,” *Networks*, vol. 50, no. 2, pp. 136–145, 2007.

[63] Pamulety, T. C. and Pillai, V. M., “Impact of information sharing in supply chain performance,” in *Technology Systems and Management*, pp. 327–332, Springer, 2011.

[64] Park, J., Gordon, P., II, M., James, E., and Richardson, H. W., "The state-by-state economic impacts of the 2002 shutdown of the los angeles–long beach ports," *Growth and change*, vol. 39, no. 4, pp. 548–572, 2008.

[65] Parmenter, D., *Key performance indicators (KPI): developing, implementing, and using winning KPIs.* John Wiley & Sons, 2010.

[66] Pettit, S. and Beresford, A., "Critical success factors in the context of humanitarian aid supply chains," *International Journal of Physical Distribution & Logistics Management*, vol. 39, no. 6, pp. 450–468, 2009.

[67] Programme, U. W. F., "Operational document so 200358: Construction and management of the wfp humanitarian logistics base at djibouti port," 2012.

[68] Programme, U. W. F., "Food aid flows 2012," tech. rep., UN World Food Programme, Rome, Italy, 2013.

[69] Robards, P. A. and Gates, W. R., "Applying Two-sided Matching Processes to the United States Navy," *Masters Thesis, Naval Postgraduate School*, no. March, 2001.

[70] Rongier, C., Galasso, F., Lauras, M., and Gourc, D., "A method to define a performance indicator system for the control of a crisis," in *8th International Conference of Modelling and Simulation MOSIM 2010*, 2010.

[71] Rongier, C., Lauras, M., Galasso, F., and Gourc, D., "Towards a crisis performance-measurement system," *International Journal of Computer Integrated Manufacturing*, vol. 26, no. 11, pp. 1087–1102, 2013.

[72] Roth, A. E., "Deferred acceptance algorithms: history, theory, practice, and open questions," *International Journal of Game Theory*, vol. 36, pp. 537–569, Jan. 2008.

[73] Roth, A. E. and Peranson, E., "The redesign of the matching market for american physicians: Some engineering aspects of economic design," 2002.

[74] Roth, A. E., Rothblum, U. G., and Vande Vate, J. H., "Stable matchings, optimal assignments, and linear programming," *Mathematics of Operations Research*, vol. 18, no. 4, pp. 803–828, 1993.

[75] Roth, A. E., Sönmez, T., and Utku Ünver, M., "Pairwise kidney exchange," *Journal of Economic Theory*, vol. 125, pp. 151–188, Dec. 2005.

[76] Roth, A. E. and Sotomayor, M., *Two-sided matching: A study in game-theoretic modeling and analysis*. No. 18, Cambridge University Press, 1992.

[77] Roth, A. E. and Rothblum, U. G., "Truncation strategies in matching markets - in search of advice for participants," *Econometrica*, vol. 67, no. 1, pp. 21–43, 1999.

[78] Sandwell, C., "A qualitative study exploring the challenges of humanitarian organisations," *Journal of Humanitarian Logistics and Supply Chain Management*, vol. 1, no. 2, pp. 132–150, 2011.

[79] Scholten, K., Scott, P. S., and Fynes, B., "(le) agility in humanitarian aid (ngo) supply chains," *International Journal of Physical Distribution & Logistics Management*, vol. 40, no. 8/9, pp. 623–635, 2010.

[80] Shapley, L. and Scarf, H., "On cores and indivisibility," *Journal of Mathematical Economics*, no. 1, pp. 23–37, 1974.

[81] Sheu, J.-B., "An emergency logistics distribution approach for quick response to urgent relief demand in disasters," *Transportation Research Part E: Logistics and Transportation Review*, vol. 43, no. 6, pp. 687–709, 2007.

[82] Smith, J., Mohring, F., and Link, D., "Making ports more resilient," *InterAction Monthly Developments Magazine*, vol. 31, no. 11, 2013. Publication status: Published.

[83] Stidham Jr, S., "Analysis, design, and control of queueing systems," *Operations Research*, vol. 50, no. 1, pp. 197–216, 2002.

[84] Stringer, E. T., *Action research*. Sage, 2013.

[85] Thomas, A., "White paper: Humanitarian logistics: Enabling disaster response," 2003.

[86] UN World Food Programme, *Emergency Field Operations Pocketbook*, 2002.

[87] Van der Laan, E., De Brito, M., and Vergunst, D., "Performance measurement in humanitarian supply chains," *International journal of risk assessment and management*, vol. 13, no. 1, pp. 22–45, 2009.

[88] Wouters, M., "A developmental approach to performance measures: Results from a longitudinal case study," *European Management Journal*, vol. 27, no. 1, pp. 64–78, 2009.

[89] Wyatt, J., "Scorecards, dashboards, and kpis keys to integrated performance measurement.," *Healthcare financial management: journal of the Healthcare Financial Management Association*, vol. 58, no. 2, pp. 76–80, 2004.

[90] Yang, W., Giampapa, J., and Sycara, K., "Two-Sided Matching for the U . S . Navy Detailing Process with," *Technical Report*, no. November, 2003.

[91] Yin, R. K., *Case study research: Design and methods*, vol. 5. sage, 2009.

[92] Zhang, J., Dong, M., and Frank Chen, F., "A bottleneck steiner tree based multi-objective location model and intelligent optimization of emergency logistics systems," *Robotics and Computer-Integrated Manufacturing*, vol. 29, no. 3, pp. 48–55, 2013.