LAND LEVELING USING OPTIMAL EARTHMOVING VEHICLE ROUTING

HOWARD D. McINVALE

Captain, United States Army

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science
in
Industrial and Systems Engineering

Pat Koelling, Chairman

Ebru Bish

Ron Wakefield

April 22, 2002
Blacksburg, Virginia

Keywords: Discrete Optimization, Heuristics, Local Search Algorithms, Vehicle Routing Problems, Traveling Salesman Problem, Shortest Route Cut-Fill Problem, Generalized Hill Climbing Algorithms

LAND LEVELING USING OPTIMAL EARTHMOVING VEHICLE ROUTING

Howard D. McInvale

(ABSTRACT)

This thesis presents new solution approaches for land leveling, using optimal earthmoving vehicle routing. It addresses the Shortest Route Cut and Fill Problem (SRCFP) developed by Henderson, Vaughan, Wakefield and Jacobson [2000]. The SRCFP is a discrete optimization search problem, proven to be NP-hard. The SRCFP describes the process of reshaping terrain through a series of cuts and fills. This process is commonly done when leveling land for building homes, parking lots, etc. The model used to represent this natural system is a variation of the Traveling Salesman Problem. The model is designed to limit the time needed to operate expensive, earthmoving vehicles. The model finds a vehicle route that minimizes the total time required to travel between cut and fill locations while leveling the site. An optimal route is a route requiring the least amount of travel time for an individual earthmoving vehicle.

This research addresses the SRCFP by evaluating minimum function values across an unknown response surface. The result is a cost estimating strategy that provides construction planners a strategy for contouring terrain as cheaply as possible. Other applications of this research include rapid runway repair, and robotic vehicle routing.

Acknowledgements

I wish to acknowledge our Creator, from whom all blessings flow, for the capacity and opportunity to learn and grow as creations. I also acknowledge the U.S. Military for sponsoring my advanced degree pursuits. I want to acknowledge my advisor, Dr. C. Patrick Koelling, for his superb mentorship, clear guidance and charming sense of humor. I also acknowledge committee members, Dr. Ron Wakefield and Dr. Ebru Bish, whose supportive expertise and thoughtful insights helped shape the direction of this research project.

I acknowledge the university, the Industrial and Systems Engineering Department, and their staff for their professionalism and reputation as the 8th-ranked program in the nation. I applaud their support for graduate students, and am grateful for the dedicated resources provided to me in the Virginia Tech Simulation and Optimization Laboratory.

I want to thank LTC Darrall Henderson for getting me involved in this project and for creating the existing model on which this project is based. I want to thank ENS Eric Shea and CPT Tracy Mann for their creative ideas and personal assistance. Finally, I want to thank my wife, Amy, for her encouragement, and my boys, Jackson and Joseph, for proving that it is indeed possible to get really excited about hauling dirt!

Dedication

This thesis is dedicated to Amy.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 SHORTEST ROUTE CUT AND FILL PROBLEM (SRCFP)

The Shortest Route Cut and Fill Problem (SRCFP) is a new method for formulating a capacitated, vehicle routing problem with pick-ups and deliveries. It seeks a route that minimizes the time required to negotiate a solution path. The SRCFP has various applications in distribution systems. An example is rapid runway repair. The concepts of SRCFP assist in optimizing vehicle routes from material stockpiles to repair sites. The application explored in this research is land leveling by an earthmoving vehicle.

The Shortest Route Cut and Fill Problem (SRCFP) finds a vehicle route that minimizes the total travel time between cut and fill locations, while leveling a construction project site. An optimal route is a route requiring the least amount of travel time for an individual earthmoving vehicle. Because of the exponentially increasing number of feasible solutions, enumerating and comparing all possible solutions to the SRCFP is an infeasible task. Therefore, it is necessary to construct efficient and effective optimization algorithms to identify optimal/near-optimal routes for construction project planners. Formulating the SRCFP as a discrete optimization problem and addressing it with local search solution strategies is the approach described in this research.

1.2 CRITICAL DEFINITIONS

The following definitions will assist in understanding the model. A construction project *site* is a plot of land with existing contours that must be modified to facilitate construction. A *cut* location is a location on the construction site with excess material and a *fill* location is a location requiring material. A *unit load* is the volume of earth (soil) that an individual vehicle can carry in one trip. *Field level* is the desired elevation and is achieved when the site's elevations are balanced. *Relative elevations* are height differences with respect to the desired elevation. A *route* is the path that a vehicle follows in order to visit every unit cut and unit fill location until the terrain modifications are complete. The *total time* is the length of time to negotiate the route plus the time spent at each cut and fill location. A *local search algorithm* is an iterative solution method, driven by a set of programming policies. Four local search algorithms are utilized in this research. The *simulated annealing*, *threshold accepting*, *Pure Local Search*, and the *Monte Carlo Search* methods will be individually discussed later.

1.3 MOTIVATION FOR CURRENT RESEARCH

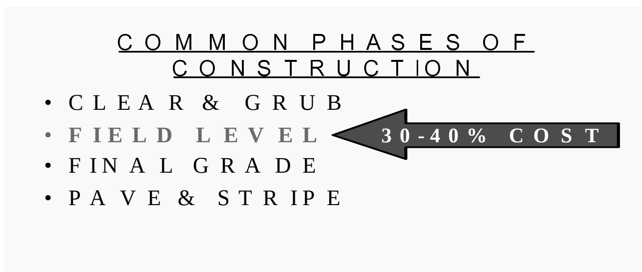


Figure 1.1 Common Phases of Construction

Construction projects often require earth transformations that involve moving large amounts of earth from point A to point B. Hauling excavated earth, from surplus (cut) locations to deficit (fill) locations, is performed by earthmoving vehicles that are expensive to operate and maintain. Since field leveling is often the most expensive portion of a construction job, minimizing the time required to operate these vehicles is cost effective.

Equipment costs significantly affect the profitability of any project. Research literature from the University of Washington’s Department of Construction Management found that the equipment cost ranks second only to labor cost in terms of uncertainty. Furthermore, land leveling estimates are inexact and usually based on historical data. Specifically, a senior field person estimates based on past experience. Because this expensive phase of many construction projects is one of the least researched, this project is an opportunity to gain more certainty in a currently unscientific process. Significant strides can be made to improve the accuracy by which field leveling costs are estimated.

These more accurate estimates improve the strategies of construction planners when they bid on projects.

1.4 PROBLEM STATEMENT AND RESEARCH OBJECTIVES

The Shortest Route Cut and Fill Problem (SRCFP) was developed by Henderson et al. [2000]. The SRCFP problem is a discrete optimization search problem, proven to be NP-hard. It describes the process of leveling terrain through a series of cuts and fills. This research addresses the SRCFP by evaluating minimum function values across an unknown response surface. Various local search techniques are used to identify minimum function values. The result of this approach is a cost estimating strategy that provides construction planners a strategy for contouring terrain as cheaply as possible.

The model used to represent this natural system is a variation of the Traveling Salesman Problem. The model finds a vehicle route that minimizes the total time required to travel between cut and fill locations while leveling the site. An optimal route is a route requiring the least amount of travel time for an individual earthmoving vehicle.

The goal of this research is to build an effective model for estimating the cost of field leveling terrain. This is accomplished by modifying the existing SRCFP model to reflect the most significant environmental features for land leveling by dump truck. Then, use experiments to examine the ability of the model to effectively incorporate these more realistic attributes. These attributes are soil variety, contoured terrain (slope), and obstacle avoidance. The experiments evaluate total route times for an earthmoving vehicle and generate a corresponding route path for the vehicle to follow. This model cannot determine optimal routing strategies. However, it can suggest reasonably good estimates in a reasonable amount of time, where *reasonable* is defined in terms of the users needs. Success for this research is a resulting model that serves as an effective tool in estimating the leveling costs for a construction application. Such a model would also provide beneficial insights into the optimization strategies for other vehicle routing applications.

CHAPTER 2: LITERATURE REVIEW

Research advances in discrete optimization search problems have evolved steadily since the 1930s. The review of the work done in this area indicated that complex routing problems, particularly asymmetric, capacitated or large instance problems were more commonly studied as computational advances made them feasible to study. The opening line to Julia Robinson's 1949 paper, "On the Hamiltonian Game (A Traveling Salesman Problem)," begins with the sentence: "The purpose of this note is to give a method for solving a problem related to the traveling salesman problem." Although the problem was apparently well known at that time, there does not appear to be any earlier written reference to the TSP. Early papers used several variations of the name *traveling salesman problem*. Robinson [1949] is the earliest reference that uses the term "traveling salesman problem" in the context of mathematical optimization. Dantzig, Fulkerson, and Johnson [1954] referred to the "traveling-salesman problem." Heller [1954] used "travelling salesman's problem," and Morton and Land [1955] preferred "the travelling salesman problem" (originally called it the laundry van problem). They were each addressing the same general problem and by the mid-1960s it was simply called the *traveling salesman problem (TSP)*.

The earliest work in this area was Menger [1932]. It addressed The Messenger Problem, a task of finding, for a finite number of points whose pair wise distances are known, the shortest path connecting the points. Robinson [1949] sought to find the shortest route for a salesman starting from Washington, visiting all the state capitals and then returning to Washington. By the 1950's solution methods began to receive greater emphasis because of the efforts of Dantzig, Fulkerson and Johnson [1954]. Their efforts included a solution of a 49-city TSP using linear-programming methods named after them.

Robacker [1955] performed by hand computational tests on the Dantzig-Fulkerson-Johnson method's solution for 10 instances of a 9-city TSP. The average time to work one instance was about 3 hours and was the benchmark for the next few years. No one attempted to verify larger instances until 1958, when Croes [1958] solved the

Dantzig-Fulkerson-Johnson 42-city example in 70 hours by hand. Also that year, Bock [1958] tested the Dantzig-Fulkerson-Johnson algorithm on Robacker's 9-city examples using an IBM 650 computer. This was the first use of computers to address this category of problem.

The model discussed in this paper is a variation of an Asymmetric Traveling Salesman Problem. The asymmetric condition on the TSP provides for scenarios in which the cost to travel between two points can depend on the direction of travel. In the land leveling application, the time it takes to go uphill from A to B can be longer than the time to go downhill from B to A. The first reference of an Asymmetric TSP was Heller [1954]. Heller's follow-up paper in 1955 discussed neighbor relations as they pertained to the ATSP. Heller [1955] and Flood [1956] were the earliest references to heuristic solution methods for ATSPs. Kuhn [1955] furthered Heller's work with a complete description of the 5-city asymmetric TSP. Morton and Land [1955] used Kuhn and Heller's developments to apply the ATSP to vehicle routing. It addressed the problem in the context of a daily one-van laundry service.

If you consider Dantzig-Fulkerson-Johnson's unique innovations as a "mouse trap," then the research in this area during the 1960's centered on "trying to build a better mouse trap." The focus of virtually all the 1960s literature in this area took what Dantzig, Fulkerson, and Johnson did, and tried to do it more and more efficiently on larger and larger problem instances. Miller, Tucker, Zemlin [1960] used an integer-programming formulation and a Gomory cutting-plane algorithm on a 10-city example. Bellman [1962] introduced the concept of dynamic programming algorithms to address the TSP, and Gonzales [1962] solved a 10-city instance using dynamic programming and an IBM 1620 computer. Held and Karp [1962] used dynamic programming to solve a 13-city instance on an IBM 7090 computer. Held and Karp also introduced a heuristic algorithm that found the optimal solution to the 42-city Dantzig-Fulkerson-Johnson example 2 out of 5 times.

As time passed, innovations continued. Little, Murty, Sweeney, and Karel [1963] were the first use of the branch-and-bound method, applying it to a 30-city ATSP. Karg and Thompson [1964] used a cheapest-insertion heuristic on a 57-city instance using data from a road atlas. Lin [1965] was an important paper in heuristic method innovation. It defined a 3-opt neighbor strategy and addressed problem instances as large as 105-cities. Reiter and Sherman [1965] used a local search algorithm on Karg and Thompson's 57-city example. Their results were comparable to Karg and Thompson's results for the cheapest-insertion heuristic. Martin [1966] outlined cutting-plane algorithms for the TSP and asserted that programming advances would reduce the sizes of models and number of iterations significantly. As the 1960's ended, it was clear that the improvements in computers would be the greatest single factor in advances in this area.

The impact of more powerful computing is evidenced by Held and Karp's research. In 1970, they introduced two methods that used minimum spanning trees to address TSPs. Held and Karp [1970] reports that neither performed well, but they intended to use intense computational evaluation to identify and correct the problems. By 1971, they did just that. Held and Karp [1971] solved the 42-city instance of Dantzig, Fulkerson, and Johnson [1954], the 57-city instance of Karg and Thompson [1964], and a 64-city random Euclidean instance. These computational results were easily the best reported up to that time. Automation advances also benefited from Hong [1972]. Hong's work was described as the most significant contribution to the linear programming approach to solving TSPs since the original paper of Dantzig, Fulkerson, and Johnson [1954]. Hong [1972] went a long way towards automating Dantzig, Fulkerson, and Johnson's method.

Computing advances fueled continued pursuits of discovery. Miliotis [1976] used a linear programming package written by Land and Powell [1973] to implement a branch-and-cut algorithm for the TSP, using subtour inequalities. Bazaraa and Goode [1977] used the dual formulation approach as a solution method for the TSP and ATSP. Miliotis [1978] combined subtour cuts and Gomory cuts to address previously solved 42-city, 48-city, and 57-city examples more efficiently. Land [1979] introduced column generation to handle the greater number edges present in larger instances and tested it on instances up to 100-cities. Crowder and Padberg [1980] contributed greatly through their use of an IBM 370 integer-programming solver to solve the 318-city instance addressed by heuristics in Lin and Kernighan [1973]. This 318-city instance remained as the largest TSP solved until 1987.

Following Crowder and Padberg's contribution, the applied aspects of the research became more frequent in the literature. People began to try and capitalize on the uses of these solvable and sufficiently large problem instances. Carpaneto, Fischetti and Toth [1988] effectively illustrated issues related to upper and lower bound analysis for TSPs. Rhee [1994] analyzed the boundary effects on the TSP, by treating the expected length of the shortest tour as a Poisson process. Both of these works also stressed the need to consider the sensitivity of system parameters.

The first work with direct application to the Shortest Route Cut and Fill Problem emerged with Anily and Hassin [1992]. Their "Swapping Problem" was a shipping application involving a single, capacitated vehicle, where the shortest route was sought. Anily and Mosheiov [1994] addressed single, capacitated vehicle TSPs with backhauls and deliveries, using heuristic solution methods. Anily and Bramel [1999] analyzed the problem of single product moves by a capacitated vehicle from pickup to delivery points. They used approximation algorithms with no routing restrictions on small instance sizes. Glover, Gutin, Yeo and Zverovich [1999] offered solution strategies to similar ATSPs using classical tour construction algorithms. Their approaches were complex and their results suggested that local search algorithms might offer a comparable, simpler approach.

Henderson et al. [2000] pursued such an approach. Their paper, "Optimal Earthmoving Vehicle Routes Using Local Search Algorithms," spawned the research in this thesis. It outlines three local search algorithms applied to a variation of the Traveling Salesman Problem (TSP). The model is capacitated, with pickups and deliveries, but unlike previous researchers, Henderson incorporates the concept of restricted routing. The purpose of this restriction is to appropriately reflect the system of earthmoving whereby a single load of earth must be deposited before another load can be acquired. Their paper evaluated up to 194 points (97 cuts, 97 fills).

Four solution techniques are evaluated in this research. Two are methods used in Henderson et al. [2000]. All four methods are local search algorithms that can be formulated using the generalized hill climbing algorithm framework (GHC) explained in Jacobson, Sullivan and Johnson [1998]. General hill climbing algorithms use an inner and an outer loop structure with general forms of the acceptance function, the neighborhood function, and the neighboring solution generating function. The objective of these algorithms is to find the best possible solution using a limited amount of computing resources. The *simulated annealing* method in Fleischer [1995] and the *threshold accepting* method in Dueck and Scheuer [1990], as well as, the *pure local search* and *monte carlo search* methods are explored in Henderson [2001] as efficient ways to find solutions that are good enough in a reasonable amount of time.

Today, broad-based experimental comparisons continue. Papers such as Cirasella, Johnson, McGeoch and Zhang [2001] use modern heuristics and offer sound experimental comparisons. However, no other papers addressed the ATSP described in Anily and Bramel [1994] and [1999] quite like Henderson. et al. [2000]. The unique quality of restricted routing persists. The model requires the dump truck to move to a place with excess dirt and load, then move to a place requiring dirt and dump it, in that order, over and over again until the terrain is level. This routing nuance does not appear in previous literature. The most similar application to date is by Anily and Bramel [1999]. They consider a capacitated ATSP with pick-ups and deliveries, but do not restrict the routing to alternate between pick-up and delivery locations.

No previous research used the Threshold Accepting method to evaluate solutions for this particular application of the SRCFP. Because of this, results using the Threshold Accepting technique will be interesting to compare with results from other local search algorithms applied by Henderson et al. [2000]. Furthermore, incorporating the Threshold Accepting solution technique and the restricted routing model modification allows for more insightful research into of the construction application of the Shortest Route Cut and Fill Problem. It is also an exciting opportunity to explore a fresh, unique twist to a highly studied, classical problem.

CHAPTER 3: METHODOLOGY

OVERVIEW

This research uses algorithms from the local search class of heuristics to address the SRCFP. Travel in these types of problems has been traditionally classified into one of three types of journeys: internal journeys, cross cordon journeys and through journeys. This research looks at internal journeys in a single area of terrain requiring field leveling.

The system in this research is modeled as a variation of the Asymmetric Traveling Salesman Problem (ATSP). The ATSP is a well-known NP-hard problem that has numerous important applications, including scheduling, facilities management and the construction application of this research.

The general formulation for an ATSP, on *m* cities, is:

$$x_{ij} = \begin{cases} 1, \text{ if } i \rightarrow j \text{ is along the route} \\ 0, \text{ otherwise} \end{cases}$$

$$\min \sum_{j=1}^{m} \sum_{i=1}^{m} c_{ij} x_{ij}$$

$$\text{s.t.} \sum_{j=1}^{m} x_{ij} = 1 \qquad \text{for } i = 1, \ldots, m$$

$$\sum_{i=1}^{m} x_{ij} = 1 \qquad \text{for } j = 1, \ldots, m$$

$$\sum_{i \in K} \sum_{j \in K} x_{ij} \leq |K| - 1 \quad \text{for all } K \subset \{1, \ldots, m\}$$

$$x_{ij} = 0 \text{ or } 1 \qquad \text{for all } i, j$$

where K is any non-empty subset of the cities $1,\ldots,m$. The cost c_{ij} is allowed to be different from the cost c_{ji}, thus making the problem asymmetric.

Three conditions are imposed on the ATSP to formulate the model used in this research. These conditions are vehicle capacity, pick-up and delivery requirements and restricted routing. Each of these will be described in detail later. To evaluate this formulation of the model, this research employs four popular search algorithms. A brief description of them and their strategies will assist in understanding the model. The first section of this chapter contains the necessary information to understand these algorithms.

3.1 UNDERSTANDING THE SOLUTION ALGORITHMS

The original formulation of the Cut-Fill Model is introduced by Henderson et al. [2001] and referred to, throughout this paper, as Cut-Fill Model 1 (CF1). The enhanced formulation is called Cut-Fill Model 2 (CF2) and is the model studied in this research. Henderson et al. [2001] addressed the SRCFP with Cut-Fill Model 1 (CF1), using the simulated annealing and monte carlo search methods to generate solutions. Cut-Fill Model 2 (CF2) uses four algorithms from the local search class of heuristics: simulated annealing and monte carlo search, plus the pure local search and threshold accepting techniques. Here is a brief explanation of the four algorithms and how they work. The simulated annealing method is used for a baseline model description and the other methods are described by how they differ from simulated annealing.

To describe the implementation of simulated annealing, the following definitions are needed. Let Ω be the *solution space* (i.e., the set of all possible solutions). Let $f: \Omega \rightarrow R$ be an *objective function* defined on the solution space. The goal is to find a global minima, ω^*, (i.e., $\omega^* \in \Omega$ such that $f(\omega) \geq f(\omega^*)$ for all $\omega \in \Omega$). The objective function must be bounded to ensure that ω^* exists. Define $\eta(\omega)$ to be the *neighborhood function* for $\omega \in \Omega$. Therefore, associated with every solution, $\omega \in \Omega$, are neighboring solutions, $\eta(\omega)$, that can be reached in a single iteration of a local search algorithm.

Simulated annealing is an algorithm used to address difficult discrete optimization problems. Simulated annealing allows for the escape from local optima, with the possibility of reaching a global optimum, by allowing for the possibility of uphill moves. Simulated annealing starts with an initial solution $\omega \in \Omega$. A neighboring solution $\omega' \in \eta(\omega)$ is then generated using some specified rule. Subsequent solutions will always be accepted if $f(\omega') < f(\omega)$, (ie. new objective function value is less). Furthermore, if $f(\omega') > f(\omega)$, then ω' is also accepted as the current solution with probability $e^{-[f(\omega') - f(\omega)]/T}$, where T is a temperature parameter that is typically non-increasing at each iteration. The process is repeated until a specified number of iterations are completed. Inferior solutions can be temporarily accepted while seeking a more minimum solution. At the end of the loop, the best iteration cost is retained as the best cost to date and the loop is repeated after updating the cooling schedule. The cooling schedule, {T(k)}, is updated by multiplying the previous temperature parameter by an increment multiplier. For the simulated annealing, the initial temperature, $T(0) = .2(C+F)M$ where C = number of unit cut locations, F = number of unit fill locations and $M = \mathrm{Max}\{\mathbf{D}(l_i,l_j),\ i,j = 1,2,\ldots,m\}$.

The Pure Local Search, Monte Carlo Search and Threshold Accepting methods can be described as special cases of simulated annealing. In particular, if T = 0, then only improving moves are accepted and simulated annealing reduces to Pure Local Search. If $T = +\infty$, and $\eta(\omega) = \Omega$ for all $\omega \in \Omega$, simulated annealing reduces to Monte Carlo Search. For threshold accepting, $T = T(k)\beta$, where β represents an acceptance threshold value.

The algorithms in this research were formulated using the generalized hill climbing algorithm framework. All generalized hill climbing algorithms have the same basic structure, but can be tailored to specific instances of a problem by changing the neighborhood function and the hill climbing random variables (which is used to accept or reject a sub-optimal move). Here's how the GHC works.

Generalized Hill Climbing Algorithms

Set the outer loop counter bound K and the inner loop counter bounds N(k), k = 1,2,…,K
Define a set of hill climbing (random) variables $R_k: \Omega \times \Omega \rightarrow \Re \cup \{-\infty,+\infty\}$, k = 1,2,…,K
Set the iteration indices i = 0, k = n = 1
Select an initial solution $\omega(0) \in \Omega$
  Repeat while $k \le K$
  Repeat while $n \le N(k)$
    Generate a solution $\omega \in \eta(\omega(i))$
    Calculate $\delta(\omega(i),\omega) = f(\omega)-f(\omega(i))$
    If $R_k(\omega(i),\omega) \ge \delta(\omega(i),\omega)$, then $\omega(i+1) \leftarrow \omega$ (accept improving or hill climbing moves)
    If $R_k(\omega(i),\omega) < \delta(\omega(i),\omega)$, then $\omega(i+1) \leftarrow \omega(i)$ (reject hill climbing moves)
    $n \leftarrow n+1$, $i \leftarrow i+1$
    Until n = N(k)
    $n \leftarrow 1$, $k \leftarrow k+1$
  Until k = K

Figure 3.1 Generalized Hill Climbing Algorithms

The neighborhood function η implemented for all experiments is defined as follows. For all routes $\boldsymbol{R} = (c_1, f_1, c_2, f_2, \ldots c_m, f_m)$, the neighbors of $\boldsymbol{R}$, $\eta(\boldsymbol{R})$, are defined by considering any $c', c'' \in \{c_1, c_2, \ldots, c_m\}$, $c' \neq c''$ or any $f', f'' \in \{f_1, f_2, \ldots, f_m\}$, $f' \neq f''$and reversing the sequence of locations between them. The result is that cuts and fills stay together in particular neighborhoods. Therefore $\eta(\boldsymbol{R}) = \{\boldsymbol{R}' \in \Omega : \boldsymbol{R}' = (c_1, f_1, c_2, f_2, \ldots, c_{i-1}, f_{i-1}, c_j, f_{j-1}, c_{j-1}, \ldots, f_{i+1}, c_{i+1}, f_i, c_i, f_j, c_{j+1}, f_{j+1}, \ldots, c_m, f_m)$, for all $i,j = 1, \ldots, m$, $i < j\} \cup \{\boldsymbol{R}' \in \Omega : \boldsymbol{R}' = (c_1, f_1, c_2, f_2, \ldots, c_{i-1}, f_{i-1}, c_i, f_j, c_j, f_{j-1}, c_{j-1}, \ldots, f_{i+1}, c_{i+1}, f_i, c_{j+1}, f_{j+1}, \ldots, c_m, f_m)$, for all $i,j = 1, \ldots, m$, $i < j\}$. To generate these neighbors at each iteration of a local search algorithm, c' or f' is generated uniformly over the set $\{l^+_1, l^+_2, \ldots, l^+_m\}$ or $\{l^-_1, l^-_2, \ldots, l^-_m\}$, respectively, and c'' or f'' is generated uniformly over the set $\{\{l^+_1, l^+_2, \ldots, l^+_m\} | \{c'\}\}$or $\{\{l^-_1, l^-_2, \ldots, l^-_m\} | \{f'\}\}$, respectively.

The four algorithms in this research are applied using the GHC framework. The formulations were obtained from Henderson et al (2000). The following chart highlights key differences in the four formulations. R_k represents the hill climbing random variable.

Method 1: Pure Local Search: $R_k = 0$

Method 2: Monte Carlo Search: $R_k = +\infty$

Method 3: Simulated Annealing: $R_k = -T(k)\ln(u)$

Method 4: Threshold Accepting: $R_k = T(k)\beta$

Figure 3.2 Summary Chart for Local Search Techniques

Pure Local Search sets $R_k(\omega(i),\omega) = 0$, $\omega(i) \in \Omega$, $\omega \in \eta(\omega(i))$, for iteration k = 1, 2,…. Therefore, $P\{R_k(\omega(i),\omega) \geq \delta(\omega(i),\omega)\} = 0$ for $\omega(i) \in \Omega$, $\omega \in \eta(\omega(i))$ such that $\delta(\omega(i),\omega) > 0$.

Monte Carlo Search sets $\eta(\omega) = \Omega$ for all $\omega \in \Omega$, and $R_k = +\infty$ for all iterations k = 1,2,… and assumes that the neighbors are generated uniformly.

Threshold Accepting sets $R_k(\omega(i),\omega) = T_k\beta$, $\omega(i) \in \Omega$, $\omega \in \eta(\omega(i))$, for iteration k, where t_k typically approaches zero as k approaches infinity (i.e., $\lim_{k \to +\infty} t_k = 0$), and $0 \leq \beta \leq 1$.

Simulated Annealing sets $R_k(\omega(i),\omega) = -T_k\ln(u)$, $\omega(i) \in \Omega$, $\omega \in \eta(\omega(i))$, for iteration k, where the cooling schedule is updated by ln(u), which represents the natural log of an uniform [0,1] random number.

MODEL COMPARISON

CF1	CF2
•UNIT SQUARE METHOD	•MAP METHOD
•USES n x n CARTESIAN GRID	•USES SELECT DATA POINTS
•SLOPE ALWAYS ZERO	•TIME ADJUSTED FOR SLOPES
•SOIL ALWAYS IDEAL	•TIME ADJUSTED FOR SOIL
•EVALUATES WHOLE GRID	•ONLY POINTS OF INTEREST
•ASSUMES 100% NAVIGABILITY	•HANDLES OFF-LIMITS AREAS
•USES 3 SEARCH METHODS	•USES 4 SEARCH METHODS
•SAMPLE DATA ONLY	•SAMPLE AND FIELD DATA
•SET VEHICLE CAPACITY	•HAS 5 VEHICLE TYPES
•COST = TOTAL DISTANCE	•COST = TOTAL TIME
•SEEKS SHORTEST ROUTE	•SEEKS FASTEST ROUTE

Figure 3.3 CF1 and CF2 Model Comparisons

3.2 CUT-FILL MODEL 1

CUT-FILL1 (CF1) is the original model addressing the SRCFP and the model that this thesis builds upon. It formulates and addresses the Traveling Salesman Problem (TSP) as a discrete optimization problem using local search strategies.

Shortest Route Cut and Fill Problem (SRCFP)

Given: a set of unit cut locations, $L^+ = \{l^+_1, l^+_2, \ldots, l^+_m\}$, and a set of unit fill locations, $L^- = \{l^-_1, l^-_2, \ldots, l^-_m\}$, and a symmetric matrix D containing the cost of traveling between the cut and fill locations in terms of distance.

Find: a *route* $R = (c_1, f_1, \ldots, c_m, f_m)$, $c_i \in L^+$, i = 1, 2, …,m, $f_i \in L^-$, i = 1, 2, …,m, such that $g(R) = \sum_{i=1}^{m-1} [D(c_i,f_i) + D(f_i,c_{i+1})] + D(c_m,f_m) + D(f_m,c_1)$ is minimized

CF1 addresses the SRCFP as a TSP with three additional conditions. First, the earth-moving vehicle is capacitated. Secondly, every solution path requires pick-ups and deliveries. CF1 also appropriately restricts the routing to where the path must alternate between cut locations and fill locations. As mentioned in chapter 2, this is the unique wrinkle in the problem. The construction application requires restricted routing, because a full dump truck *must* empty before reloading. Contrast this with the postman or dry cleaning delivery guys who can drop-off or pick-up at several homes in a row to save time. They can also pick-up from and deliver to the same home in one visit. No routing restriction makes them repeatedly stop and pick-up, then stop and deliver, home-by-home, like land leveling requires. Thankfully, their routing is not restricted; mail is already slow enough.

CF1 is formulated for a simplified problem that does not account for many of the physical attributes of earthmoving operations. It assumes straight-line distances, no off-limit areas, and ideal surface conditions. It also assumes slope in all directions to be zero. However, CF1 evaluates total distance, such that the shortest path wins, and it illustrates effective uses of local search algorithms to address the SRCFP. The limitations of the CF1 model created the opportunity for the continued research explored in this thesis.

3.3 CUT-FILL MODEL 2

ASYMMETRIC **C**APACITATED **T**RAVELING **S**ALESMAN **P**ROBLEM w/ **P**ICK-UPs & **D**ELIVERIES & **R**ESTRICTED **R**OUTING

Figure 3.4 Technical Title for the CF2 Model

CUT-FILL2 (CF2) is modeled as an Asymmetric Traveling Salesman Problem (ATSP). Additionally, it is a capacitated model, requires pick-ups and deliveries and imposes restricted routing. CUT-FILL2 (CF2) is an enhancement of the CF1 model. CF2 better represents the physical conditions on a construction site. The CF2 model accounts for the aspects of the CF1 model, as well as soil and slope effects and obstacle avoidance. Also, rather than evaluating travel distance, CF2 evaluates travel time. CF2 integrates the manufacturer's specifications for the earthmoving vehicle and evaluates the total time to negotiate the solution path. Here is the formulation for CF2.

Given: a set of unit cut locations, $L^{+} = \{l_1^{+}, l_2^{+}, ..., l_m^{+}\}$, and a set of unit fill locations, $L^{-} = \{l_1^{-}, l_2^{-}, ..., l_m^{-}\}$, and an integrated time matrix ***T***, comprised of:

[travel-time matrix ***D***, soil factor matrix ***S***, slope factor matrix ***M***], that represents the travel times between all the cut and fill locations.

Find: a route ***R*** = (c_1, f_1,…, c_m, f_m), where $c_i \in L^{+}$, for i = 1, 2, ..., m, and $f_i \in L^{-}$, for i = 1, 2, ..., m, such that g(***R***) = $\sum_{i=1}^{m-1} \{\boldsymbol{T}(c_i,f_i) + \boldsymbol{T}(f_i,c_{i+1})\} + \boldsymbol{T}(c_m,f_m) + \boldsymbol{T}(f_m,c_1)$ is minimized.

Both CF1 and CF2 evaluate function values across unknown response surfaces, as they seek near-optimal solutions. However, another distinction between the models is the way that input values are determined. The CF1 approach is a unit square method whose input values lie along uniform grid locations. The CF2 approach can evaluate values across a non-uniform grid space. This flexibility allows a user to handpick the points of interest and level only those. The CF1 model affixed an appropriately sized n x n grid to the terrain that accommodated all the points requiring leveling. This resulted in much larger problem sizes that took longer to solve. The CF2 model's ability to be tailored to the needs of the user is it's strongest attribute. The next section desceibes the CUT-FILL2 Model (CF2).

3.3.1 CF2 MODEL DESCRIPTION

System *N* is the process of leveling terrain through a series of cuts and fills. Since it is not feasible to represent all physical realities that impact *N*, the parameters and measurable elements of the system were carefully identified to capture the most important aspects of the system.

$$N = \{\Omega, A\}$$

where,
Ω = SOLUTION SPACE, set of all possible states for the system.
A = ABSTRACTION, things selected to be measured.

Appropriately and thoroughly structuring the CF2 model resulted in a model that performed well and accurately simulated the system it represented. The CF2 model is structured to reflect the way the leveling process behaves. Here are the justifications for the constraints on the model, followed by descriptions of the parameters and measureable elements of the model.

<u>JUSTIFICATION</u>:

ASYMETRIC: time to travel from (node1, node2) not always equal for (node2, node1)

CAPACITATED: there is a capacity of one truck full for every move

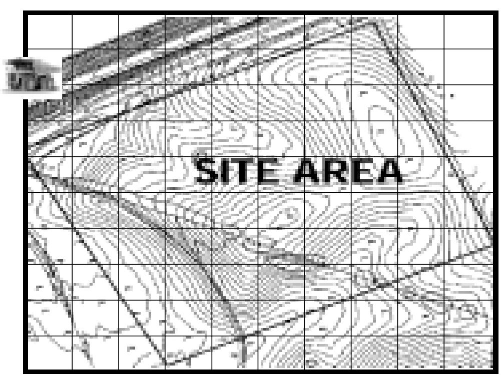
TRAVELING **S**ALESMAN **P**ROBLEM: each route forms a Hamiltonian circuit

PICK-UPs & **D**ELIVERIES: all routes require both cuts and fills

RESTRICTED **R**OUTING: ●, ●, ●, ●, routing restriction (must dump before filling)

PARAMETERS OF THE MODEL

1. **COORDINATE LOCATIONS**



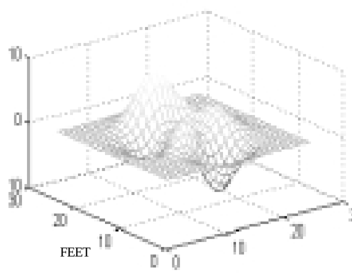
The topography data is taken from locations within the site area, using 8-digit grid coordinates. All cuts and fills occur at one of these coordinate locations.

2. **EARTHMOVING VEHICLE**

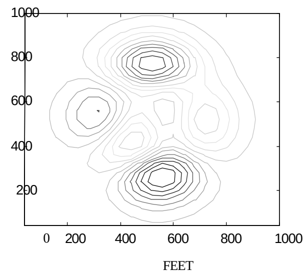
The individual earthmoving vehicle in the system, with the manufacturer's specifications for operating and maintaining it.

3. **DIMENSIONS OF SITE AREA**

3-D Representation



Contour Representation



The dimensions of the site area include the length and width of the site area, and the relative elevations at each coordinate location.

4. **DESIRED ELEVATION**

Determined by the designer, and is referred to as the "field level" elevation. The result of the model is for all of the coordinate locations to have this elevation.

5. **SURFACE CONDITIONS**

The type of soil or material the earthmoving vehicle moves across, including any weather effects that alter the surface.

MEASUREABLE ELEMENTS OF THE MODEL

1. **EUCLIDEAN DISTANCES**

   Basis for all calculations in model

Straight-line distance from each data point to every other data point. Distance is converted into a travel time, *t*, from every point to every other point.

2. **SOIL COMPOSITION**

   Divided into terrain categories:

CATEGORY	EXAMPLE
"Go"	Packed Gravel
"Slow Go"	Loose Sand
"Very Slow Go"	Wet Clay
"No Go"	Off-Limits Areas

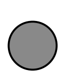
3. **RELATIVE ELEVATION**

Example:

DESIRED ELEVATION: 2060 ft
ACTUAL ELEVATION: 2063 ft
RELATIVE ELEVATION: +3 ft

How much above or below the spot elevation is, with respect to the desired elevation (field level). Note: 1 cut or 1 fill = 1 foot change in elevation.

+3

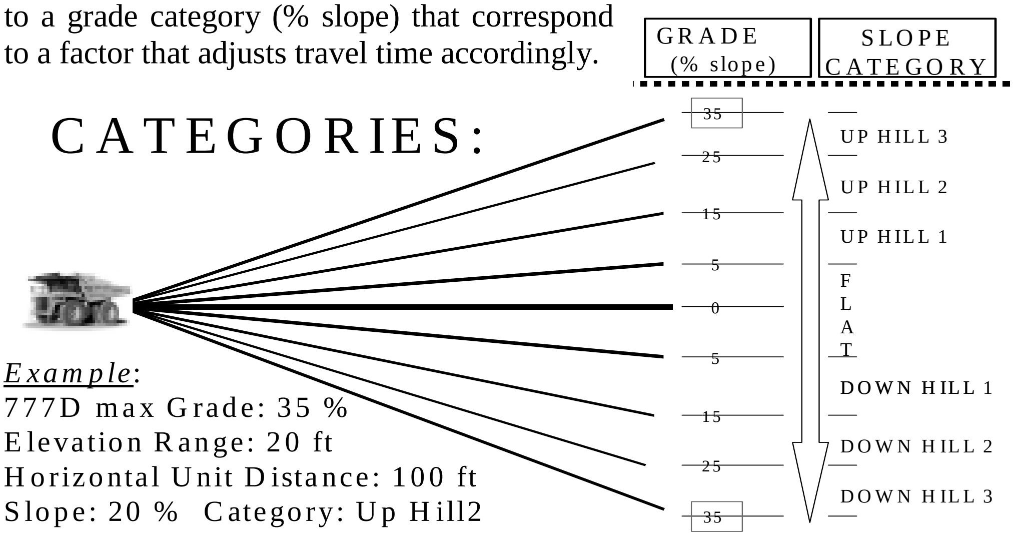


= REQUIRES 3 CUTS DURING ROUTE PATH TO BE LEVEL

4. **SLOPE**

Elevation changes between points are assigned to a grade category (% slope) that correspond to a factor that adjusts travel time accordingly.

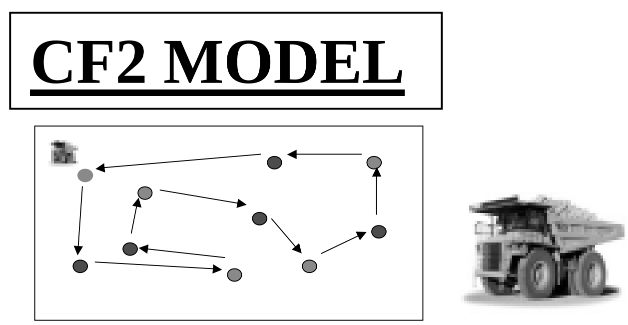
CATEGORIES:



Example:
777D max Grade: 35 %
Elevation Range: 20 ft
Horizontal Unit Distance: 100 ft
Slope: 20 %   Category: Up Hill2

3.3.2 MODEL ILLUSTRATION

THE MODEL…



PROBLEM: Given a set of "p" points, $P = \{p_1, p_2, \ldots, p_n\}$, and a Time matrix T, comprised of:
[distance matrix D, soil matrix S, slope matrix M],
that represents the travel times between points in P.

MODEL: Find a *Hamiltonian circuit*, $H = (p_1, p_2, \ldots, p_n)$, such that $f(H) = [\sum_{i=1}^{n-1} T(p_i, p_{i+1})] + T(p_n, p_1)$ is minimized.

Figure 3.5
The CF2 Model Formulation

Each possible route forms a Hamiltonian circuit that visits each cut / fill location once and returns to the starting point. The number of TSP locations, *n*, will require multiple visits to the same physical locations whenever the relative elevation is > 1. For every *n* total locations, the CF2 model has (n – 1)! number of feasible routes. Here are examples of two out of (9)! possible routes in a case with 10 total cuts / fills.

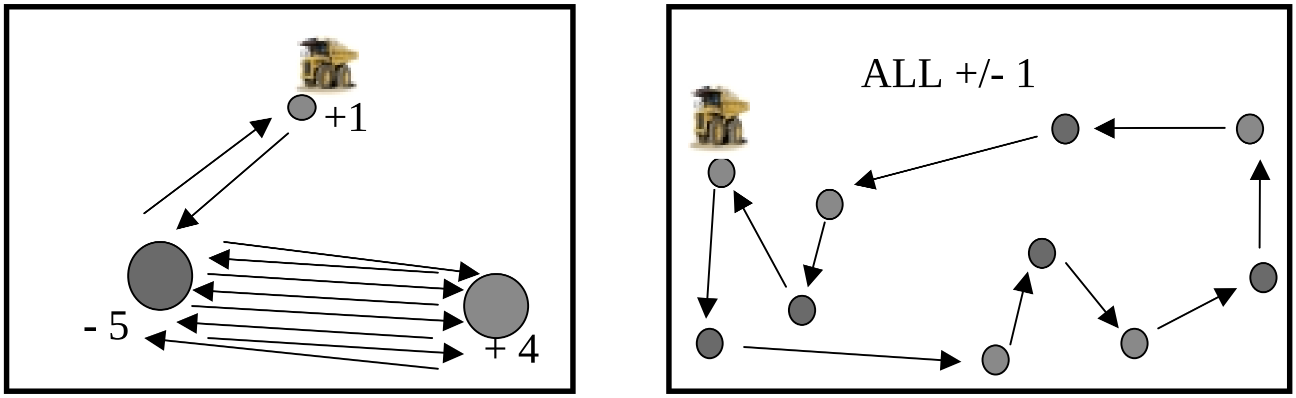


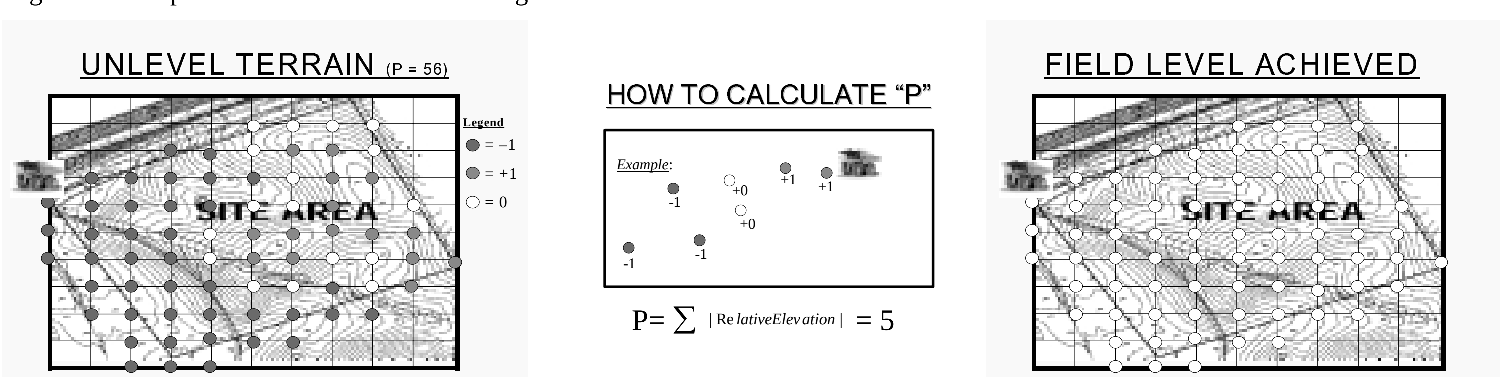
Figure 3.6 Illustrations of a Hamiltonian Circuit

THE PRODUCT…

Figure 3.7 Example of a Finished Product

THE PROCESS…

Figure 3.8 Graphical Illustration of the Leveling Process



3.3.3 CF2 MODEL ASSUMPTIONS

1. The physical terrain at a given data point matches what is reflected on the map.
2. The earthmoving vehicle is self-loading.
3. Vehicle is either full or empty, for every move.
4. No partial deliveries allowed. Truck fills completely, then empties completely.
5. 1 unit cut or 1 unit fill = 1 truck load
6. Slope for multiple visits is static.

3.3.4 CF2 MODEL INPUTS

The input for the model is an integrated time matrix:

$$\begin{bmatrix} Travel \\ Time \end{bmatrix} \bullet \begin{bmatrix} Soil \\ Factor \end{bmatrix} \bullet \begin{bmatrix} Slope \\ Factor \end{bmatrix} = \begin{bmatrix} Integrated \\ TimeMatrix \end{bmatrix}$$

Comprised of:

- *TRAVEL- TIME MATRIX*
  How long to travel the path?
  Euclidean distances across an ideal surface (i.e. packed gravel)

- *SOIL FACTOR MATRIX*
  How much longer due to the soil?
  Formulated as a % increase of the travel time across an ideal surface

- *SLOPE FACTOR MATRIX*
  How much longer / shorter due to slope?
  Formulated as a % increase or decrease of the travel time across an ideal surface

3.3.5 CF2 MODEL OUTPUTS

The primary model output is a total time required for the vehicle to negotiate the route. ROUTE TIME (RT) is a function of: Travel Time, Soil Factor, and Slope Factor. A complete list of output variables representing such things as best cost, worst cost and solution route path is located in Appendix B.

3.3.6 CF2 COST FUNCTION

In addition to the Route Time calculated in the model, the idle time required to load each cut and dump each fill is necessary to determine the total time the vehicle is required.

IDLE TIME (I) = ($P_{C*}I_C$) + ($P_{F*}I_F$), where C= Cut, F= fill, P= # points

Therefore, *TOTAL TIME = ROUTE TIME + IDLE TIME*

TOTAL TIME, multiplied by the rate for renting the vehicle, yields the total cost. Renting earthmoving vehicles is common in the industry since most construction companies do not own their own fleet of earth-moving equipment. The CF2 model's cost estimates serve as smart contract bids on construction projects. The model is capable of generating cost estimates that provide useful information in the other applications as well.

3.3.7 CF2 OPTIMAL SOLUTION

Since Idle Time and total cost are linearly linked to the number of cuts and fills, their values are not factored into the model. The best Route Time implies the best overall cost. Therefore, the incumbent "fastest route" solution generated by the model at the end of the specified number of replications is considered the best available solution. It may or may not be the optimal solution to the entire problem. In fact, it is almost always not the optimal solution for sufficiently large problem instances.

3.4 RESEARCH PLAN

3.4.1 Research Questions

This section states the questions of interest in this research.

- How do the CF2 model cost estimates compare with the cost estimates in CF1?
- Does the CF2 model effectively accommodate real world data?
- What parameters govern model performance?

- How does the model perform on various processing systems?
- What strategies will assist in the use and implementation of the CF2 model?
- How well do the four local search algorithms perform in addressing the SRCFP?
- How does the model perform under increased loads?
- For what size problem instances is the CF2 model best suited?

3.4.2 Research Procedure

This section outlines the procedures followed to evaluate the algorithm's ability to address the CF2 model. The following steps were followed.

1. Verify the performance of the model. Use various problem instances and each of the four local search techniques.
2. Compare the performance of the PC and Unix systems. Use an identical problem instance to monitor trends.
3. Create a systematic way to generate good field data from topography maps and incorporate the physical attributes of terrain into the data builder.
4. Apply CF2 to a select location and graphically compare before and after appearance.
5. Perform experiments on the field data to determine the parameters that drive the model’s performance.
6. Suggest strategies that can be useful in selecting which model parameters to use for given situations.
7. Identify the model’s limitations and the natural extensions for research in this area.

3.5 SUMMARY

The goal of the experiments was to test the CF2 model and assess its performance. The answers to the research questions are the outcome of the research procedure. The first phase of research ensured that the model functioned properly. Sample data was used to verify and test the model's performance. The second phase, the performances of processing systems, search techniques, and the model as a whole was evaluated. Field data were formatted using a worksheet based data generator. Microsoft Excel 2000 was used to create the algorithm's input data through a worksheet based data generator created for this model. The data builder was the means by which field data were converted into an n x n matrix that MATLAB could interpret and process. Topography attributes of coordinate locations, spot elevations and soil categories, taken from maps, were the inputs required for the data builder. This information will be taken from maps. The data builder's output is an n x n matrix worksheet containing the integrated travel times from each point, to every other point. This output worksheet served as the necessary input information for the MATLAB code to run experiments on the field data.

The evaluation process included experimental runs performed on the data using four heuristic search algorithms. Using a variation of the effective, existing algorithm designed to address CF1, the CF2 model was executed in MATLAB 12. The four heuristics applied in the model were Local Search, Monte Carlo Search, Simulated Annealing and Threshold Accepting.

Finally, trials were performed on PC and Unix systems to provide execution insights and performance comparisons for these two processing systems. Excel was used to store and display input and output information. Graphs and visual aids were created in Excel and MATLAB. MATLAB 12 and Microsoft Excel 2000 are sufficiently updated versions of software for these purposes. A Dell PentiumIII PC (clock speed 666 Mhz with 512 Mb of RAM) and an Unix *SunBlade 1000* Model 750 system (clock speed 900 Mhz with 512Mb of RAM) were used to perform the experiments. Both are located in the Industrial and Systems Engineering's Simulation and Optimization laboratory.

CHAPTER 4: EXPERIMENTS AND RESULTS

4.1 MODEL VERIFICATION

Computer-generated sample data was used to verify that the CF1 model functioned properly. Sample data was used for all of the runs previous to this research, therefore its performance had to be well established before integrating the additional attributes for the CF2 model. A list of all of the runs performed is located in Appendix A. The runs appear by date with the name, size, inner/outer loops and replications listed. All of the runs were made in MATLAB. PC sample runs were titled, “dougshovel.m” and Unix sample runs were titled, “DOUGSHOVEL.m.” Several modifications to the code were made to streamline the models performance. With each modification, verification runs were made for identical size instances and using each of the four search techniques. Answers were compared to previously attained answers to ensure they matched. The resulting code demonstrated identical answers to those from the CF1 model, only slightly faster. This was a result of cleaning up how temporary memory and output files were handled. Anticipating heavy workloads to be placed on the system, the simplest possible model would perform in an easy to analyze way. This was especially important considering that no benchmarks existed for the CF2 model. An accurate, effective but simple code provided less opportunity for errors that might be hard to identify without benchmarks. The revised and final version of the MATLAB program was re-titled, ‘Thesiscode.m’ and remained the only code used in the remaining experiments.

The next verification was done on real data sets. Experiments were performed on various size problem instances using all four search techniques. All runs on real data were titled, ‘Thesiscode.m’ whether on PC or Unix systems and are also in Appendix A. Trends were used to determine that the real data runs were performing well. The answers matched between systems, and the run times for each search technique ranked the same as they had during sample run trials. With this satisfactory evidence that the model performed consistently, the next step was to execute the planned experiments.

4.2 SAMPLE DATA RUNS: HEAD-TO-HEAD PC VS UNIX

The first experiment compared the performance of the PC and Unix systems, using an identical problem instance. The goal was to monitor trends and infer as to the performance of the two processing systems. The sample data used was identical to a sample data set used in experiments with the CF1 model. Note that Henderson et al. [2001] ran CF1 trials solely on PC systems, so Unix performance was unknown. The same data was tested on the same exact system to verify it. As expected, the performance was the same. The data set consisted of a distance matrix and an initial vector. The response surface was a unit square grid with each unit equal to 1. The distance matrix was size 49 x 49 and represented the point-to-point distances on the response surface. The initial vector was size 1 x 50 and represented the relative elevations at each coordinate point. The distance matrices were always n x n, and the initial vectors were always 1 x (n+1). The extra point represented an off-site location, located one unit off of the response surface. It's relative elevation was sufficient to allow the system to balance.

Total Iterations = (#*m* inner loops) x (#*k* outer loops) x (#*n* replications). The tasks performed in each loop are listed below.

Inner loops in the algorithm:

1. Generated random route paths that balanced the system.
2. Evaluated the total distance for that path.
3. Compared each iteration to previous total distance and saved the smaller as best.
4. Executed neighbor switches to allow for different routing to be evaluated.
5. Repeated for *m* loops.

Outer loops in the algorithm:

1. Logged the best cost during the previous inner loop.
2. Adjusted the cooling schedule for the search technique and restarted inner loop.
3. Updated the best cost with each loop.
4. Repeated for *k* loops.

Replications in the algorithm:

1. Logged the best cost for the entire replication
2. Repeated the sequence for *n* replications.
3. Outputed the variables of interest.

Historical data existed from previous CF1 trials of 10,000 iterations and 20,000 iterations. The 10,000 iteration trials were used for comparison. CF1 and CF2 resulted in identical answers of 191 minutes for total route time. This confirmed the parity in the two models. Once this was established, the evaluation criterion was the processing time for the two systems. The number of iterations was increased to evaluate both the PC and Unix systems. The results are presented in Table 4.1.

Table 4.1 Table Data for Head-To-Head System Trials

Sample Data Runs: Head-to-Head PC vs UNIX				
PC Run Titles	# Trials	Avg Time (min)	Cost Range(hrs)	Avg (Std Dev)
PC7-S-100-10-10	10	1.15	181-200	191 (4.0)
PC7-S-100-10-100	10	11.87	180-202	192 (4.0)
PC7-S-100-10-500	10	51.78	173-205	191 (4.4)
PC7-S-100-10-1000	10	54.18	176-204	190 (4.4)
PC7-S-100-10-10000	3	568.80	169-205	188 (4.5)
Unix Run Titles	# Trials	Avg Time (min)	Cost Range(hrs)	Avg (Std Dev)
U7-S-100-10-10	10	1.33	181-196	191 (4.0)
U7-S-100-10-100	10	12.73	179-199	192 (4.1)
U7-S-100-10-500	10	37.95	179-203	191 (4.4)
U7-S-100-10-1000	10	44.50	176-203	190 (4.5)
U7-S-100-10-10000	3	447.63	169-207	189 (4.4)

The run titles identify each set of trials. The notation for the titles is **Cn-A-I-O-R**, where

C: the system that performed the trials

n: the n x n size grid from which the data was drawn

A: the search algorithm applied

[Pure Local (L); Monte Carlo (M); Simulated Annealing (S); Threshold Accepting(T)]

I: the number of inner loops performed

O: the number of outer loops performed

R: the number of replications performed

The results of the head-to-head comparisons indicated that both systems arrived at the same solution for similar scenarios. As the number of replications increased, the answers for average cost improved, as expected. Both systems found the same average cost in all cases except the 10,000 replications case. The averages were slightly varied because only three answers were averaged. This was evidence that more than three answers should be generated for consistent averages in future experiments.

A conclusion from this experiment was that running a sufficient number of replications averages out the slight deviations in the cost ranges. The lower bound, on the cost ranges in the table above, is the single best solution found anywhere during the simulation. As expected, the lower bounds decreased as the total number of processed iterations increased. The longer the search algorithm looked, the better answer it was able to find. However, the cost ranges between the PC and Unix systems varied slightly. This happened because initial solutions were randomly generated in each inner loop. For each iteration, the processors used random number generation to assign new solution paths and identify the neighbors to switch. When the assignments are not identical, then different solution paths are generated that may have different total costs. The differences in the way the two processors handled these random assignments accounted for the slight variation in cost ranges. The sets of trials above labeled [100-10-10,000] demonstrated that increasing the number of replications reduces the variation. When only 3 replications were used, the average costs differed. 10 replications appeared to be a sufficient number, and future experiments would allow for this conclusion to be confirmed or denied.

The final and most interesting comparison between the PC and Unix systems was their processing times. The time to complete a trial was considered the processing time. This comparison criterion was the most useful for evaluating the two systems. When identical processes are performed on identical data, in an almost identical manner, the answers should be virtually identical. However, the time to execute this task is dependent upon the system performing the task. The head-to-head case study showed tremendous parity between the PC and Unix systems for trials in which the number of iterations was lower (< 100, 000 iterations). In fact, the PC was sometimes faster than the Unix system.

These results illustrated the effects of the computer's visual outputting on the processing time. The minimum time threshold for generating visual outputs in the PC was lower than that of the Unix system. Therefore, when the system printed graphs and variable values to the screen, the impact of the faster visual outputting caused the overall PC processing time to be faster in smaller trail sizes. Once the processing times exceeded the PC's fixed time for visual outputs, the Unix system's faster processing ability manifested in the overall processing times.

As the number of total iterations increased, the Unix system consistently finished faster. Unix was 20% faster in trials of ten million iterations, solving in under 8 hours, what the PC took over 9 hours to solve. This result suggests that the memory management in a Unix system is superior to a PC for larger trials. For smaller numbers of iterations, both systems were equally likely to finish first. For iterations greater than 100,000, the Unix system more efficiently handled the larger matrix operations and consistently performed better.

4.3 FIELD DATA RUNS: BUILDING THE DATA

Before field data could be processed, it had to be acquired. The pursuit to evaluate real world data led to the CF2 model's ability to evaluate a non-uniform grid. CF1 determined the relative locations of cut and fill sites by where they lay on a cartesian coordinate system. CF1 used a uniform, unit square grid and basic trigonometry to calculate point-to-point distance. Because land doesn't naturally contour in a grid pattern, preferred cut and fill locations aren't distributed in a grid pattern. The uniform grid technique of CF1 often resulted in the algorithm evaluating many unnecessary data points. Previously, 9 data points, distributed across an area of terrain, could require no less than a 3 x 3 grid to cover them (assuming each point fell exactly at a coordinate node). A more likely grid size required to accommodate nine data points was 10 x 10 or even more. Even if a 10 x 10 square grid could cover each point, the model would evaluate and level 100 points in order to level the 9 data points of interest. Consequently, the time needed to accomplish this would be longer.

An innovation in CF2 was the use of grid locations to calculate point-to-point distance. Simple distance formulas were created to compare longitudinal and latitudinal positions and calculate straight-line distance. The algorithm was not limited by the grid boundaries; leveling estimates were possible for anywhere in the world (provided a map of the terrain existed). This contribution to the research was a distinction between CF1 and CF2. The non-uniform grid approach provided more flexibility and specificity.

If 9 data points required leveling, only 9 data points were evaluated. The result was custom contoured terrain at individually selected locations. The time to achieve this was greatly reduced. The decision in the CF1 model, “What size grid is sufficient to provide coverage to all the points needing leveling?” was now simply, “What points need leveling and how many are there?”

While evaluating the CF2 model’s potential for accurately predicting land-leveling costs, an implied task was to acquire the data for the model to process. The method used to build the input data was called the *Mustard Seed Data Builder*. This worksheet based data generator was created for this model. This data builder was the means by which field data was converted into a processable format. The CF2 model provided the ability to input three topographic values from a map location and instantly generate an accurate travel time from that point to every other data point. The data builder was created in Microsoft Excel 2000, with one sheet accepting the raw topography data and a second sheet yeilding a final, integrated travel time matrix. An example of the input sheet for the Mustard Seed Data Builder is presented in Table 4.2.

Table 4.2 Sample Mustard Seed Data Builder Input Worksheet

VEHICLE	#
773D	*1*
775 Quarry	*2*
777D	*3*
785B	*4*
Scraper	*5*

VEHICLE # 3

Speed Data		
Loaded speed	20	mph
Empty speed	38	mph

DATA SET:

DATA16EG-3-140

Points 1-16

Mustard Seed
Data Builder

Slope Category	% Grade	Slope Factor
Max Up	35	1000000
Up Hill 3	25	1.8
Up Hill 2	15	1.5
Up Hill 1	5	1.2
Flat	-5	1
Down Hill 1	-15	0.9
Down Hill 2	-25	0.8
Down Hill 3	-35	0.7
Max Down	< -35	1000000

Soil Type	Soil Factor
Go Soil	1
Slow-Go Soil	1.3
Very Slow Soil	2
Off-Limit/No-Go	1000000

INPUT FIGURE

	x-coord	***y-coord***	***elevation***	***soil type***
Point #1	2565	9055	140	3
Point #2	2560	9060	125	3
Point #3	2552	9048	125	3
Point #4	2555	9040	125	3
Point #5	2575	9055	125	3
Point #6	2562	9070	125	3
Point #7	2550	9065	125	3
Point #8	2530	9070	125	3
Point #9	2520	9060	125	3
Point #10	2530	9055	125	3
Point #11	2540	9063	110	2
Point #12	2545	9063	110	2
Point #13	2542	9058	110	2
Point #14	2540	9045	100	2
Point #15	2545	9045	100	2
Point #16	2515	9015	125	2
Point #17	0000	0000	0	0
Point #18	0000	0000	0	0
Point #19	0000	0000	0	0
Point #20	0000	0000	0	0
Point #21	0000	0000	0	0
Point #22	0000	0000	0	0
Point #23	0000	0000	0	0
Point #24	0000	0000	0	0
Point #25	0000	0000	0	0
Point #26	0000	0000	0	0
Point #27	0000	0000	0	0
Point #28	0000	0000	0	0
Point #29	0000	0000	0	0
Point #30	0000	0000	0	0
Point #31	0000	0000	0	0
Point #32	0000	0000	0	0

The measureable elements in the system came from the raw topography data on the map sheet. The data required were grid coordinate location, spot elevation and soil type. An 8-digit grid was used because it was accurate to the nearest 10 meters. This compared to a 6-digit grid (accurate to within 100 meters) and a 10-digit grid (accurate to within 1 meter). The amount of earth necessarry to comprise a typical dump truck load was between 10m x 10m x 1 foot deep and 25m x 25m x 1 foot deep. Therefore, relocating within 10 meters of a location was acceptable and an 8-digit grid was deemed appropriate. Spot elevations were the actual feet above sea level that the map showed for each data point. The spot elevations were compared to the field level elevation to determine the relative elevations for each point. The spot elevations were also used to calculate the slope from point-to-point. Soil type was the surface material at a given point. The type of soil was important because the travel times vary along different surfaces of movement. The topography data and vehicle type were entered above. The travel times from every point to every other point were based on these entries.

Technical specifications for five earthmoving vehicle types were built into the worksheet. The Caterpillar brand earthmoving vehicles types were: 773D dump truck, 775 quarry truck, 777D dump truck, 785B dump truck and a scraper. All of the calculations in the data builder were linked to the type of vehicle used. An entry in the "vehicle#" block (Table 4.2) caused an automatic update to the worksheet. The technical specifications that applied to speed, slope and soil were updated. Adjustment factors for slope and soil were assigned to adjust the travel time for the slope and soil across which the vehicle moved. These factors were gathered from construction industry publications and technical manuals for the various vehicles. The primary source was the Caterpillar Handbook. This technical manual was furnished by the Caterpillar Company and provided the necessarry manufacturer's specifications for operating and maintaining the vehicles.

Speed, slope and soil calculations were all of the values needed to generate output in the CF2 model. The output was an n x n matrix worksheet containing the integrated travel times from each point to every other point. An example of a Mustard Seed Data Builder output page is shown in Table 4.3.

Table 4.3 Sample Mustard Seed Data Builder Output Worksheet

	Pt #1	Pt #2	Pt #3	Pt #4	Pt #5	Pt #6	Pt #7	Pt #8	Pt #9	Pt #10	Pt #11	Pt #12	Pt #13	Pt #14	Pt #15	Pt #16
Point #1	0.00	0.22	0.38	0.46	0.26	0.39	0.46	0.98	1.16	0.90	0.67	0.55	0.60	0.69	0.69	1.65
Point #2	0.15	0.00	0.37	0.53	0.41	0.26	0.29	0.81	1.03	0.78	0.52	0.39	0.47	0.64	0.55	1.64
Point #3	0.38	0.37	0.00	0.22	0.62	0.62	0.44	0.80	0.88	0.59	0.49	0.43	0.36	0.38	0.23	1.27
Point #4	0.46	0.53	0.22	0.00	0.64	0.79	0.66	1.00	1.04	0.75	0.71	0.64	0.57	0.41	0.34	1.21
Point #5	0.26	0.41	0.62	0.64	0.00	0.51	0.69	1.22	1.42	1.16	0.92	0.80	0.85	0.94	0.81	1.85
Point #6	0.39	0.26	0.62	0.79	0.51	0.00	0.33	0.82	1.11	0.91	0.59	0.47	0.60	0.86	0.78	1.86
Point #7	0.46	0.29	0.44	0.66	0.69	0.33	0.00	0.53	0.78	0.57	0.26	0.17	0.27	0.57	0.53	1.57
Point #8	0.98	0.81	0.80	1.00	1.22	0.82	0.53	0.00	0.36	0.39	0.31	0.43	0.44	0.69	0.75	1.46
Point #9	1.16	1.03	0.88	1.04	1.42	1.11	0.78	0.36	0.00	0.29	0.52	0.65	0.57	0.64	0.75	1.16
Point #10	0.90	0.78	0.59	0.75	1.16	0.91	0.57	0.39	0.29	0.00	0.33	0.44	0.32	0.44	0.46	1.10
Point #11	0.51	0.39	0.37	0.53	0.69	0.44	0.20	0.24	0.39	0.25	0.00	0.10	0.10	0.35	0.36	1.04
Point #12	0.42	0.29	0.32	0.48	0.60	0.35	0.08	0.32	0.49	0.33	0.10	0.00	0.11	0.36	0.35	1.09
Point #13	0.45	0.35	0.27	0.43	0.64	0.45	0.20	0.33	0.43	0.24	0.10	0.11	0.00	0.25	0.26	0.98
Point #14	0.52	0.48	0.19	0.30	0.70	0.64	0.43	0.52	0.48	0.22	0.35	0.36	0.25	0.00	0.10	0.75
Point #15	0.34	0.41	0.12	0.17	0.61	0.58	0.40	0.56	0.56	0.35	0.36	0.35	0.26	0.10	0.00	0.82
Point #16	1.23	1.64	0.96	0.91	1.39	1.39	1.18	1.10	0.87	0.82	1.04	1.09	0.98	0.75	0.82	0.00

Data16EG-3-140
Integrated Time Matrix

Pts: 16
Location: Egypt vic Grid 90..25..
Truck Type: 3 (777D Caterpillar Dump)
Elevation Factor: 140 (Total cuts and fills)

+20	+5	+5	+5	+5	+5	+5	+5	+5	+5	-10	-10	-10	-20	-20	+5

Data16EG-3-140a
Initial Vector of Relative Elevations

MUSTARD SEED
DATA BUILDER
OUTPUT FIGURE

Pts: 16
Location: Egypt vic Grid 90..25..
Truck Type: 3 (777D Caterpillar Dump)
Elevation Factor: 140 (Total cuts and fills)

This output matrix of adjusted travel times (size n x n), along with a relative elevations vector (size 1 x n), were the input data on which the MATLAB code ran experiments to test the data.

4.4 FIELD DATA RUNS: TESTING THE DATA

EGYPT SCENARIO

The ability to process actual field data led to the next set of experiments. The experiment was to select an area of terrain and use it as a case study to examine the CF2 model's performance. The terrain chosen for the case study was in Egypt. The specific location was a desert region in the Sinai peninsula (vicinity grid 90502550). The source of the topography data for this area came from U.S. Military map sheet 6180 published in 1990 by the Defense Mapping Agency. Previous work related visits to the Sinai allowed for lots of personal experience relocating sand and driving various wheeled-vehicles in desert surface conditions. The familiarity with the terrain would allow for "common sense checks" of the model to ensure that the results were realistic. This would not be possible for unknown terrain. There would be no way to verify results, short of actual physical experiments.

Situation: The site area was a desert river bed, known as a wadi. Most of the year, wadis are a travel route through the desert for people and vehicles. However, in the rainy season it becomes a raging river. Assume that the rainy season has just finished. The locals desire the wadi to be widened to accommodate increasing traffic through the area.

The area was 400m x 400m. The soil was comprised of two soil categories: Slow-Go soil (packed sand) and Very Slow soil (loose, dry sand). The elevation range was 95-150 feet. The lower elevations run north to south along the wadi with increasingly higher ground on the east and west banks. The northeast corner of the site area was the highest point at 150 feet. The terrain is represented in Figure 4.1.

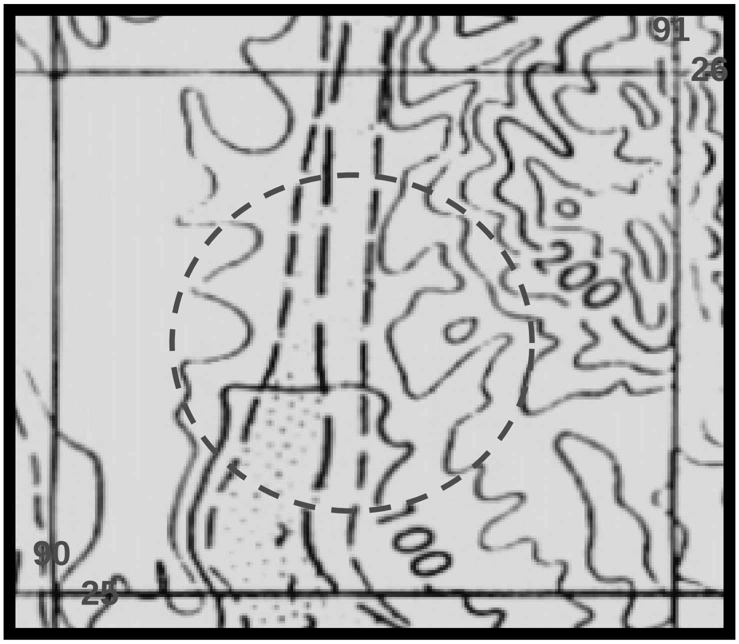


Figure 4.1 Topography Map Blow-up for Egypt Case Study

The initial set of trials were to evaluate CF2's ability to level the terrain. Initially, the experiments were executed like the sample data runs where terrain was deemed level when all relative elevations were zero. However, this seemed insufficient. A way to visualize the process seemed much more insightful. The Mustard Seed Data Builder has a capacity of 64 data points. Therfore, the first step was to create the 64-point data set graphically. The resulting 3-D and periphery views were created to better visualize the process. The goal of the experiment remained the same, but the visual aids more clearly demonstrated the results. Below are views of the terrain before any modification occurred. It will be helpful to refer to these images throughout the next few experiments to easily compare before and after appearances.

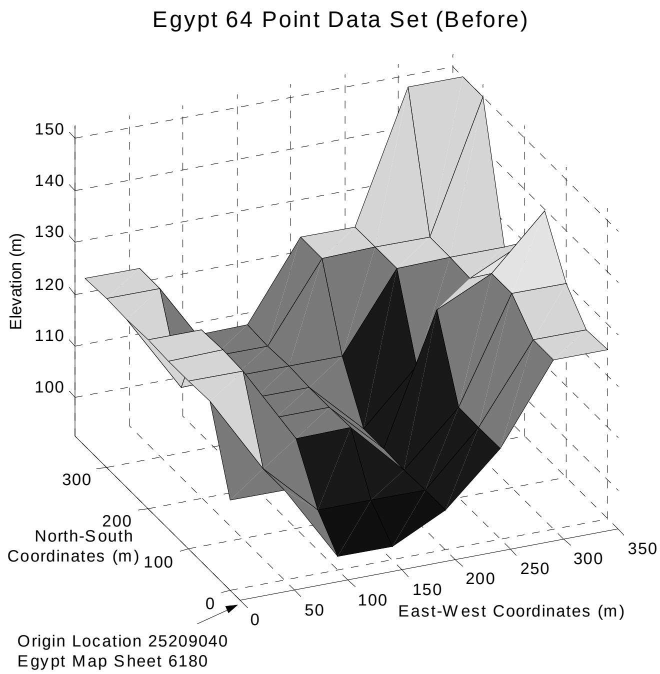
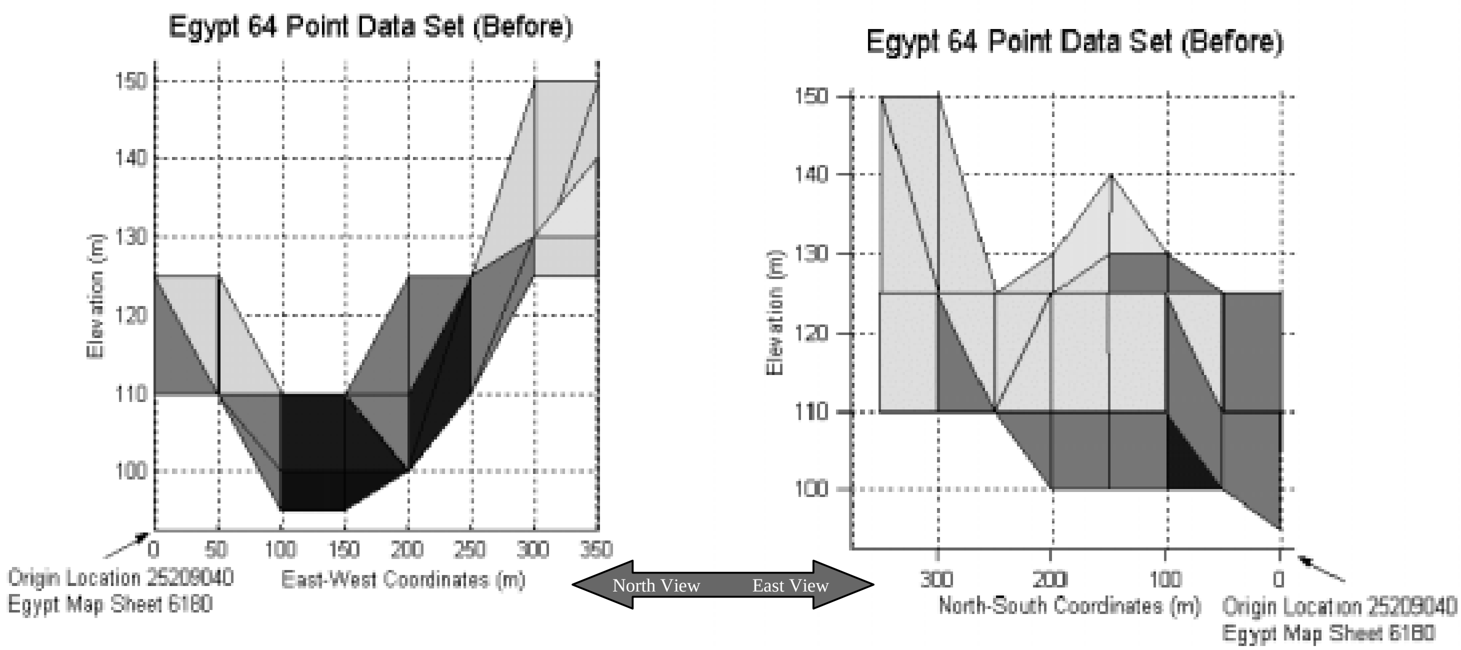


Figure 4.2 Egypt 64-Point 3-D Graphs (Before)



16-POINT TRIALS

A sequential process was followed to examine the leveling process. The trials were designed to level 16 points, then 16 different points, then 32 points, then all 64 points. Progress was monitored graphically for each set of trials. Each set of trials provided opportunities to evaluate the four search techniques in the CF2 model, and to further compare the PC and Unix system performances.

Recall that the elevation factor, *p*, was calculated using the following equation:

$$\sum |\,\mathrm{Re}\,lativeElevation\,|$$

It represents the sum of the absolute values of all of the relative elevations. In short, it is the total number of loads that must be relocated in order to balance the system. In the Egypt scenario, the problem initially had 16 data points, 30,000 iterations, and an elevation factor, p = 140. Considering that p = 140 for the 16 data points, evaluating 30,000 possible solutions was a small trial size. 139! (equal to 9.61 x 10^{238}) total number of individual solution paths existed.

The results for the two 16-point data sets were as follows:

Table 4.4 Table Data for 16-Point Experiment

Run Title	# Trials	Avg Run Time (min)	Cost Range (min)	Avg Cost / Std Dev
PC16-EG-L-3-1000-10-3	10	3.34	56-57	57 (0.2)
PC16-EG-M-3-1000-10-3	10	3.34	65-66	65 (0.3)
PC16-EG-S-3-1000-10-3	10	3.33	61-63	62 (0.2)
PC16-EG-T-3-1000-10-3	10	3.33	59-61	60 (0.4)
Run Title	# Trials	Avg Run Time (min)	Cost Range (min)	Avg Cost / (Std Dev)
U16-EG-L-3-1000-10-3	10	3.35	56-58	57 (0.3)
U16-EG-M-3-1000-10-3	10	3.31	64-66	65 (0.4)
U16-EG-S-3-1000-10-3	10	3.30	61-63	62 (0.3)
U16-EG-T-3-1000-10-3	10	3.30	59-61	60 (0.3)

**Titles are coded as follows:

[PC/ Unix & 16 Points]–[Egypt]–[Search Method]-[Vehicle Type]–[# inner loops]–[# outer loops]–[# reps]

At the completion of each trial, the relative elevations at these 16 points were zero. This indicated that the points were level. Each technique's performance is listed above. As expected, answers for each technique were the same on the PC and Unix. The average processing times were virtually equal also. The cost, in terms of travel time, is represented by the average column. The Pure Local Search technique, abbreviated [L] above, performed considerably better than the other three techniques. This is consistent with its reputation as a strong search technique, particularly for smaller problem instances. Leveling the 16 data points recountoured the terrain to appear like Figure 4.3.

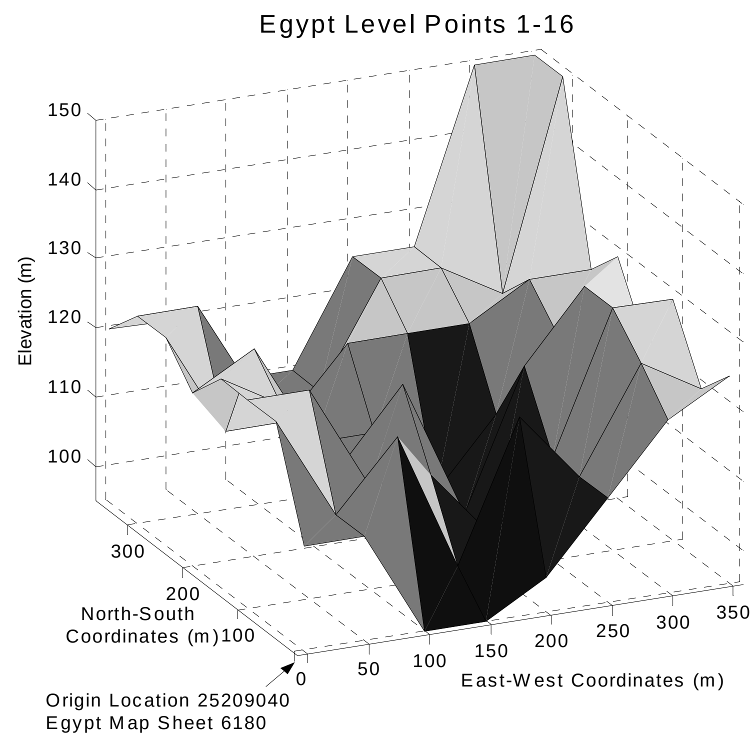
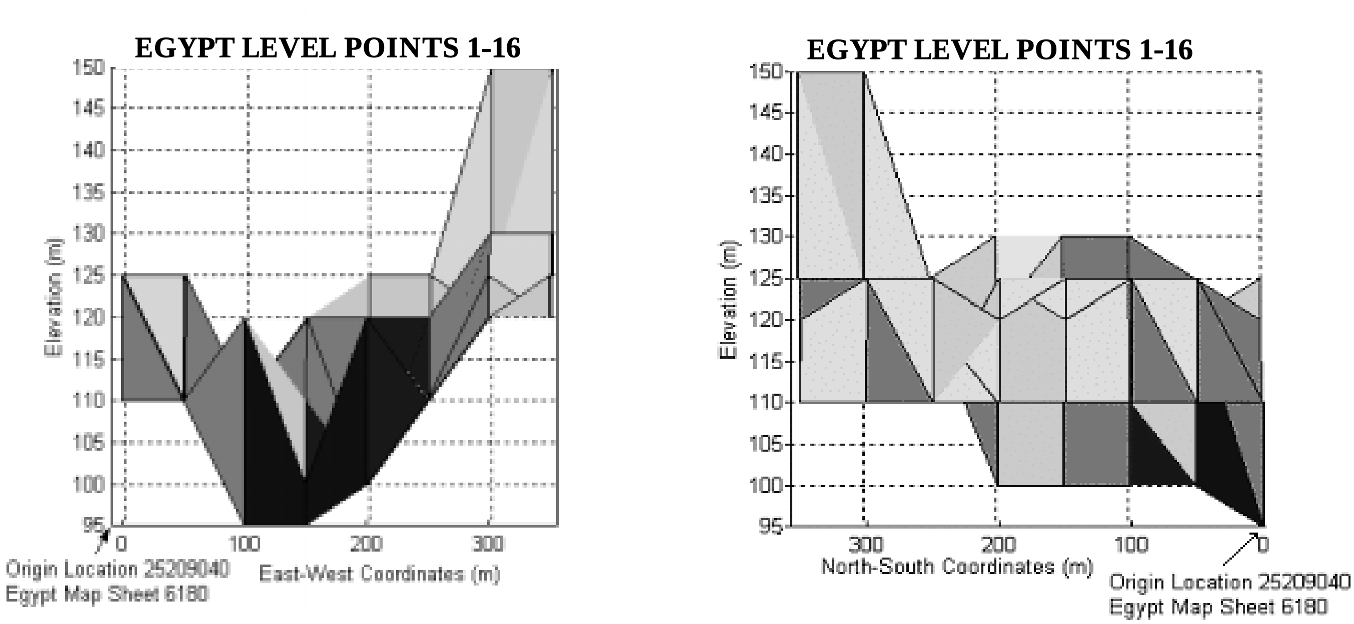


Figure 4.3 Egypt Level Points 1-16 Graphs

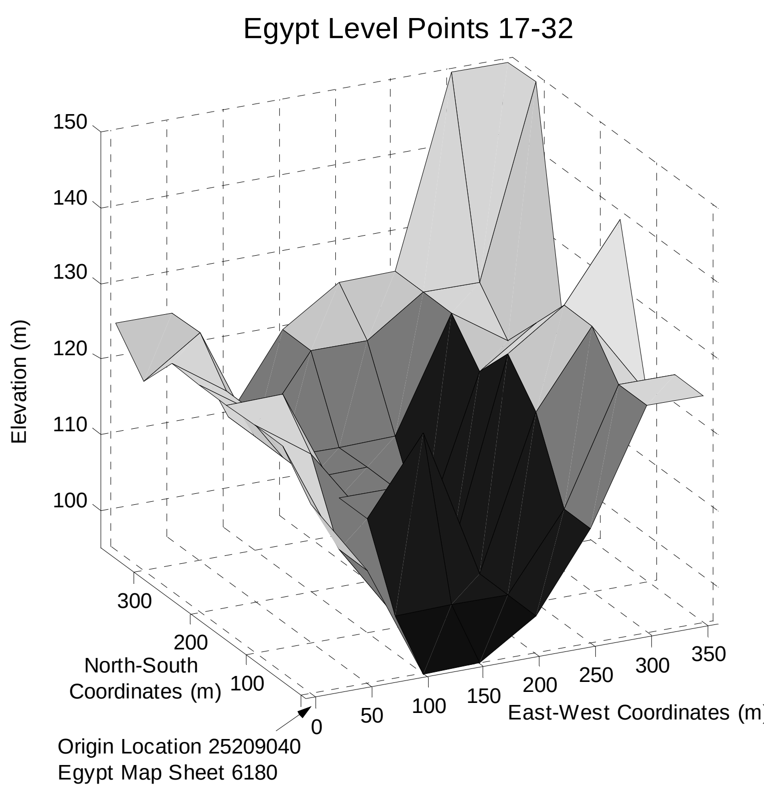


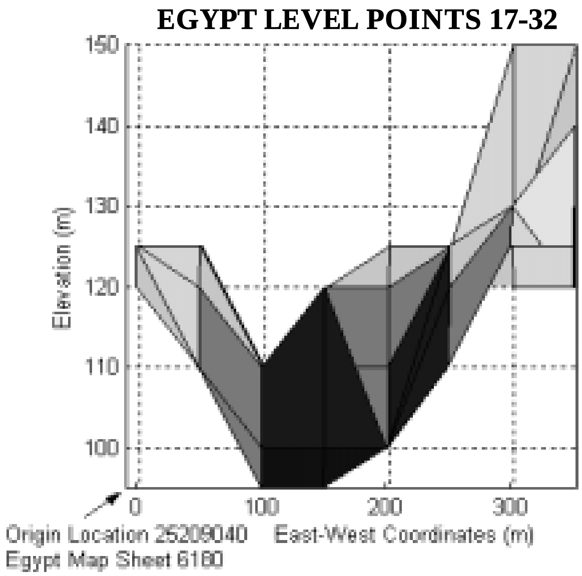
To verify that it mattered which points were selected, 16 different points in the same 400m x 400m area were selected and evaluated. The results demonstrated that the appearance indeed changes. This emphasized the importance of point selection. The model only levels what it is instructed to level. Smartly chosen points can go a longer way toward a level appearance than randomly distributed data points.

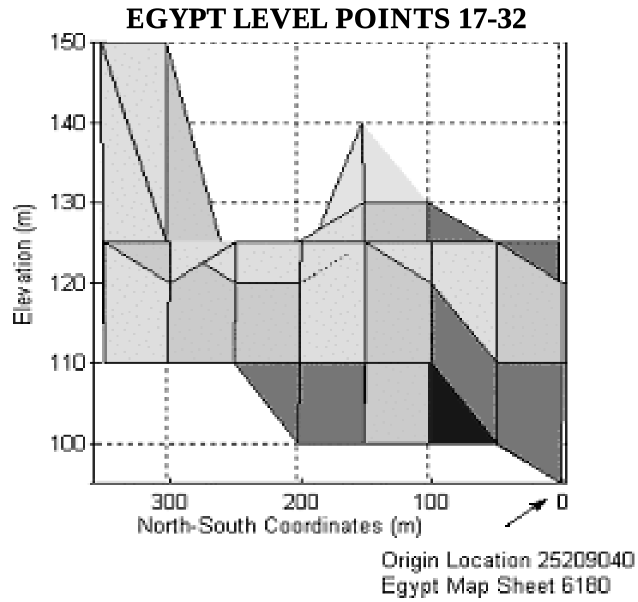
Another reason to evaluate a second set of 16 points was to compare whether two sequential leveling processes take longer than one large leveling process Additionally, a second set of 16 points would shed light on how to decide whether 16 or 32 points would make the most sense from a cost perspective. This information would be useful to make cost decisions regarding how in-depth an estimate a construction project planner could afford.

This set of trials also provided a second evaluation of the search techniques. The values for processing time and average cost differed from the previous set of trials of 16 points, but the same trends occurred. The run tables are in Appendix E. Pure Local Search performed solidly better than the other three which each performed the same. The rationale is the same as previously stated and the graphical illustration is in Figure 4.4.

Figure 4.4 Egypt Level Points 17-32 Graphs







32-POINT TRIALS

The next set of trials were for a 32-point data set with p = 280. The trials for the 32-point data set were the most comprehensive. Points of interest included:

1) Sequential vs Collective Leveling Analysis
2) PC vs Unix Performance
3) Search Technique Comparisons
4) Model Performance with Increased Loads

The trial results are in Appendix E. The graphical illustrations are in Figure 4.5.

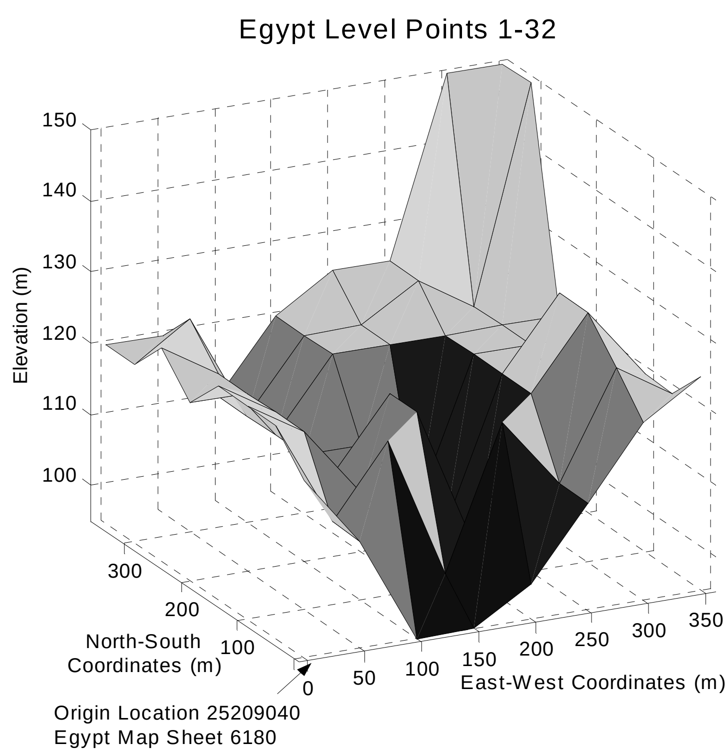
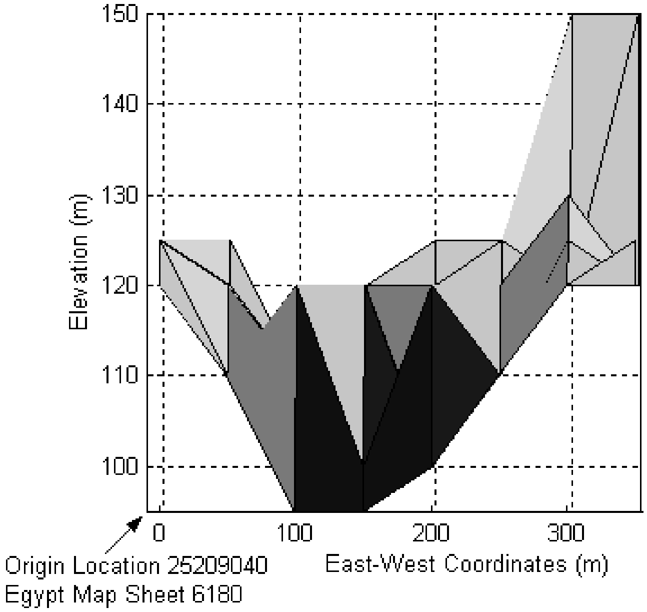
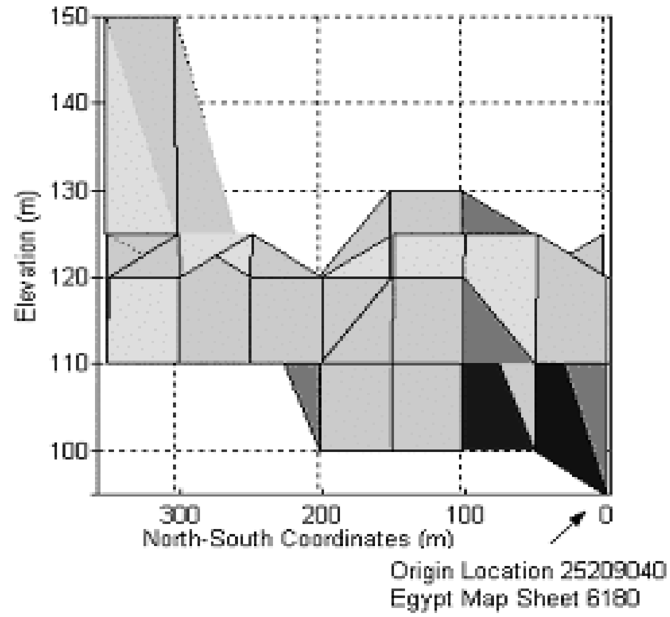


Figure 4.5 Egypt Level Points 1-32 Graphs

EGYPT LEVEL POINTS 1-32



EGYPT LEVEL POINTS 1-32



1) <u>Sequential vs Collective Leveling Analysis</u>

The CF2 model handled the 32-point data set well. The graphs clearly show that increasing the number of data points achieved a more level appearance. The cost analysis follows the phrase, “You get what you pay for.” The 16-point cost estimates were less accurate than the 32-point cost estimates. They evaluated half the points and produced a poorly leveled area. The 32-point estimates were more accurate and better leveled the terrain. However, terrain of that size will hardly be flat because 32 points is still not enough to smoothly contour the site. Considering that each load in a 777D dump truck is an area of earth that is 20m x 20m x 1foot, only about 10% of the surface will be modified. Now if 90% of the surface is already at field level then great! If not, the vehicle capacity and the number of data points can be increased to provide enough coverage to the site.

An interesting observation was made regarding time during the trials of the 32-point data set. For trials on an equal number of data points, the processing times were linearly linked to the number of iterations performed. Consider the processing time for run PC32-EG-M-3-1000-10-10. This was a PC run from the Egypt 32 data point set using a 777D dump truck. It was a set of trials of the Monte Carlo search technique for 1000 inner loops, 10 outer loops and 10 replications. The average processing times for the run was 20.38 minutes. The runs for exactly 10 times as many iterations lasted 200.60 minutes. This supports the linear relationship. However, when comparing the 16-point data sets to the 32-point set, an interesting observation emerged.

The trials on the two 16 point data sets were separately performed and tabulated. Then the two sets were combined and processed as one 32-point data set. The expectation was for 32 to take a processing time equal to 16a + 16b. The 32 data points were leveled faster for all four search techniques. On average, both the PC and Unix systems processed the sets 10% faster than leveling them separately. Separately, the two sets perform certain tasks in the algorithm that are sized dependent and some that are not. Calculating costs and finding solutions are size dependent. Performing neighbor switches and outputting variable values is not size dependent. Since the tasks that were not size dependent occurred at a constant rate, the processing times for larger problem sizes were slightly less than a 1:1 ratio to problem size. This explained the slight improvement in the processing times for 32 points. The conclusion that 32 points leveled in 10% less time than the total time for two 16-point sets is good news in that it may motivate planners to seek out the more accurate estimates found in larger data sets.

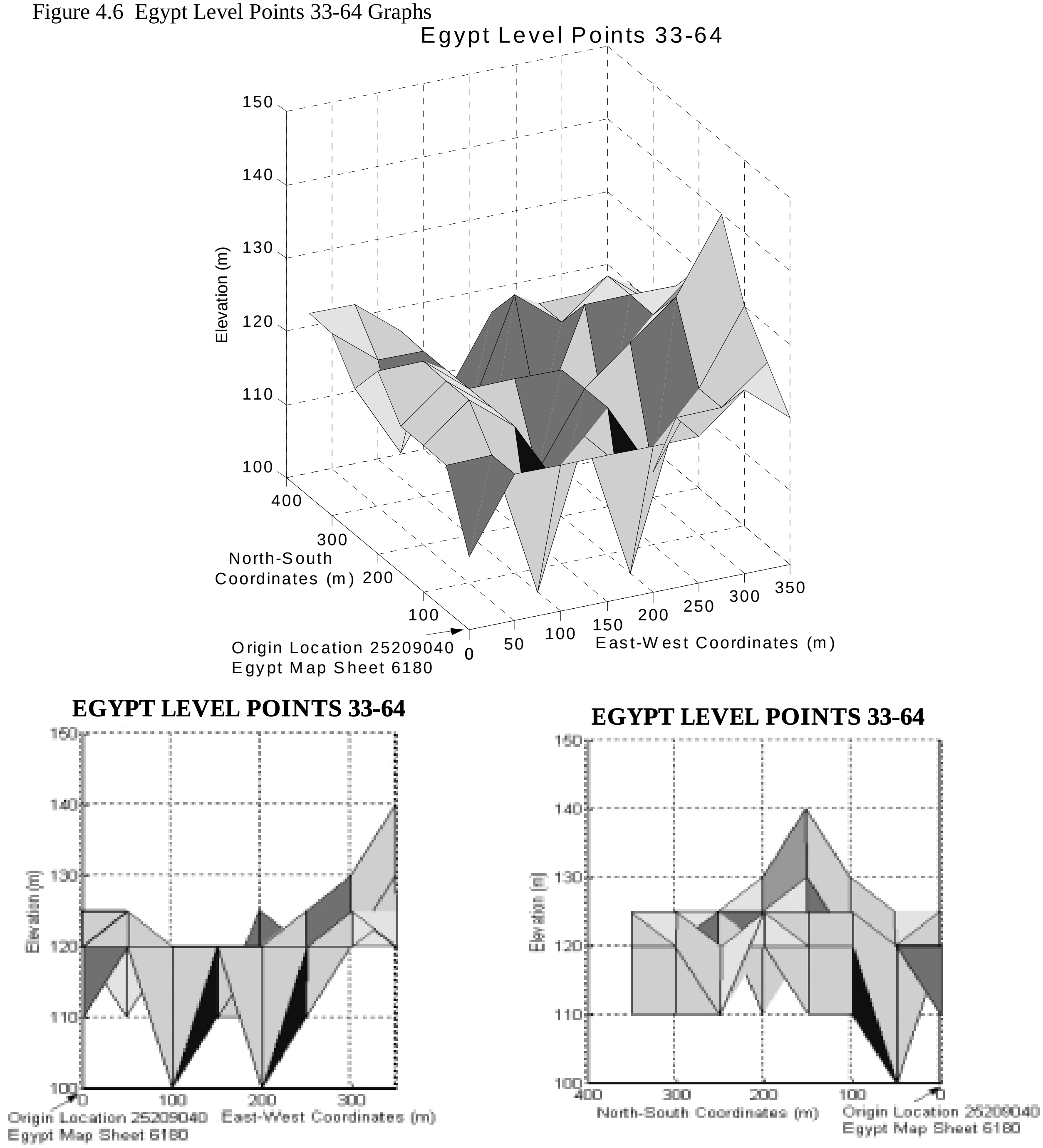
However, an important point to stress is that 32 points may not always take longer than 16 points. The elevation factor, p, also impacts the travel time because it determines the number of moves required. So, a couple of tall hills and deep holes could take weeks to level. That is why the comparison of 16 and 32 point sets was done with p= 140 for the 16 point data sets, and the p= 280 for the 32-point data sets.

2) PC vs Unix Performance

Factors that affected the run times were the fixed time requirements for the systems to produce the visual outputs to the screen (discussed in section 4.2), the variability in the way each system implemented the MATLAB and memory management efficiency. In the 32-point trials, the Unix and PC found the same values for average cost, but the Unix system consistently found them faster. This held true as iterations increased beyond 100,000 and for larger values for p. Particularly, as the number of iterations got large, the Unix system clearly outpaced the PC. The indication was that a Unix system has more efficient memory management. Each algorithm performed numerous matrix operations requiring temporary memory to be used. The Unix system appeared slightly better able to execute these operations.

The next experiment plotted a different 32-point set in the same 400m x 400m site area. Terrain appearance and the model performance were evaluated. The previously drawn conclusions for 16 and 32-point data sets were confirmed in the comparisons of the 32 and 64 point data sets. The analysis is the same and the conclusions are the same. The graphs for the second 32-point data set are shown in Figure 4.7.

Figure 4.6 Egypt Level Points 33-64 Graphs



3) Search Technique Comparisons

The results of the trials on the 32-point data sets yielded some insights about the four search strategies used in the CF2 model. The analogy that best describes the search technique's performances is auto racing. Local search and Monte Carlo search performed like cars with powerful engines and little gas tanks. Simulated Annealing and Threshold Accepting performed like fuel-efficient cars with large gas tanks. The first two performed well initially then leveled off. The last two methodically found improving solutions as the number of inner loops increased. Local search was particularly successful at finding the best average cost in trials with the fewest inner loops. In the larger trials, the simulated annealing technique was narrowly better than the threshold accepting technique, which narrowly bested the Monte Carlo search. Pure local search leveled off in the second set of trials and showed no improvement in the third, fourth and fifth sets of trials.

Other observations were made about the performances of the four search techniques. The search strategies ranked the same for both the PC and Unix systems. The standard deviations for the average cost estimates were consistent for trials with the same number of replications. The difference between the best and worst average estimates was never more than 4 minutes. This is a very small range for a construction project leveling estimate. This indicated that each method performed well for its designed purpose. In the case of more time sensitive applications, like rapid runway repair and unmanned aerial surveillance, 4 minutes may matter greatly. In some instances, every single minute may matter.

For situations in which every minute matters, note that the range of costs for all iterations for all trials was 160-171. Simulated annealing found the 171-minute route in the 30,000-iteration trial (100 inner x 10 outer x 3 reps). Simulated annealing also found the 160-minute route in the 10,000,000-iteration trial (100,000 inner x 10 outer x 10 reps). This supports the previous conclusion that increasing the number of inner loops greatly improved the accuracy. This was especially true for simulated annealing, which took full advantage of the search opportunities the longer inner loops provided.

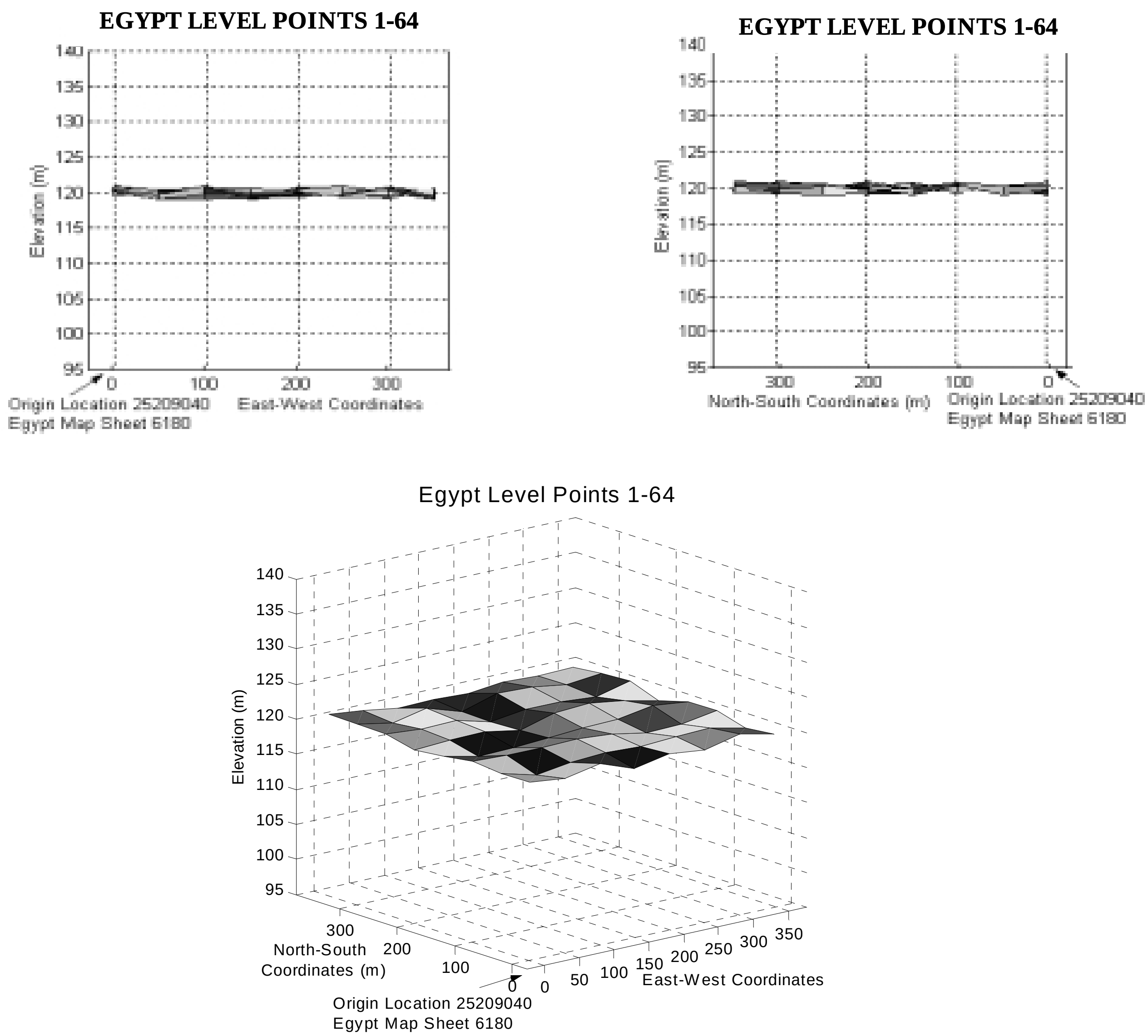
4) Model Performance with Increased Loads

The two ways used to evaluate the model's performance with increased loads on it were a set of 64-point trials for the Egypt scenario and another set of trials on data taken from terrain with extreme conditions, such as Hawaii.

64-POINT TRIALS

The final experiment from the Egypt scenario, tested the complete, 64-point data set with a p = 560 and graphed the results. The terrain, at each of the 64 points, was leveled to within 1 foot of the field elevation of 120 feet. The experimental results are graphed in Figure 4.7. Discussion of this experiment will be in Chapter 5. It will be used to illustrate some overall conclusions.

Figure 4.7 Egypt Level Points 1-64 Graphs



HAWAII ILLUSTRATION

To illustrate the CF2 model in another set of conditions, an experiment was performed on terrain on the island of Oahu, Hawaii. The purpose of this experiment was to apply the model to another scenario and demonstrate the model's performance with a different vehicle and some extreme soil and slope. The Hawaii illustration showcases additional aspects of the model that were selected for highlighting in this example. Refer to the map in Appendix F as needed.

Situation: Time is very short to establish a landing zone vicinity grid FJ07257635. Helicopters will land there to evacuate wounded soldiers who will be centralized there. The area needs to be 400m x 400m in size. Only one dump truck is available for the job. The field level elevation is 1260 feet, the median elevation in the area. A 20m x 20m area in the southwest corner is an endangered snail habitat. Do not disturb it. How soon can helicopters begin evacuating casualties?

This is a very practical example to illustrate the CF2 model's performance under increased loads. It highlights more of the features and capabilities of the CF2 model. The illustration used truck type 4 because was larger than the type 3 truck used in the Egypt case study. The type 4 truck (785B) provides 25% coverage to the 400m x 400m site area. The soil was wet, red clay throughout the site area, belonging to the Very Slow category. The slope from point-to-point was predominately in the Up Hill 3 category. The elevation factor, p, for the 64-point data set was 2500. Each relative elevation was a factor of 10 feet, so the elevation factor was scaled by a factor of 10 to allow for parallel comparisons with the Egypt scenario which had an elevation factor, $p = 240$.

The following information describes some useful output that the CF2 model generates. Specifically, the point-by-point relative elevations for each corresponding data point are shown in Table 4.5. Also, the corresponding solution route path is shown in Table 4.6, which lists the load-by-load movements to follow in order to execute the leveling job. Also illustrated is how No-Go/ Off-Limits areas are accounted for in CF2.

Table 4.5 Hawaii: Relative Elevation Assignment

Hawaii Illustration

Topography map data is 1985 Oahu Special edition 1-29EBT
Hawaii location is 400m x 400m centered vic FJ07257635 (Elevation 1360 feet)

Field Level Elevation = 1260 ft.
Initialvector = Data64H-4-250a
Integrated Time Matrix= Hawaii 64 x 64

Relative Elevation Assignments (vector size 1 x 64)

Columns 1 through 18

1	0	1	-5	-4	10	6	-4	-8	6	4	-2	-3	-7	5	-9	5	6

Columns 19 through 36

2	3	-5	3	2	-8	-9	5	6	4	1	0	1	0	1	0	-2	-3

Columns 37 through 54

-2	0	5	6	-5	-4	8	9	3	2	1	-5	6	-4	-3	-4	-3	3

Columns 55 through 64

4	-6	-8	-2	2	3	-7	-3	0	1

Example: 1) Data Point #9 has a relative elevation of -8 and requires 8 fills to achieve field level.
2) Data Point #30 has No-Go / Off-Limits soil. No solution path will include this point.
3) Data Point #59 has a relative elevation of +2 and requires 2 cuts to achieve field level.

Note: The absolute values of these numbers correspond to # of trips made to those locations.
The sequence of these trips form the Route Path illustrated on the next page.

The relative elevations determined how many total possible solution paths existed. The algorithms evaluated a very small portion of them and suggested a reasonable answer for the optimal solution path. Data points with a relative elevation of zero are one of two types of points. One type is a location whose elevation equals field level. This occurred in the CF1 model, because the CF1 model superimposed a grid on the terrain and the elevations were taken from the coordinate nodes (some of which were already level). Because the Mustard Seed Data Builder doesn't use a grid structure, this type of point is never chosen in the CF2 model. Selecting level points only wastes a data point and is not useful in improving the terrain's appearance. The second type of point is No-Go / Off-Limits points. This type of point is one that is not to be driven across. It was either selected by the user to be off-limits or the soil at that point is not navigable.

One way to use such points is to protect an environmentally sensitive area. In the Hawaii illustration, data points #30, #32, #34, and #38 were used as a perimeter to an off-limits area. The CF2 model accommodates this by inputting the field level value as the elevation at that point and assigning it the No-Go/Off Limits soil category (a factor of 1,000,000 min). A preferred solution path avoids the area and levels the remaining areas effectively. The suggested route path to level the Hawaii illustration is shown below.

Table 4.6 Hawaii: Best Solution Route Path

HAWAII Route Path for Best Solution to Date (vector size 1 x 250)

Columns 1 through 18

31 14 10 56 6 61 43 25 7 36 49 24 10 16 20 16 49 24

Columns 19 through 36

29 25 17 4 27 25 17 41 7 41 39 16 40 4 6 48 17 62

Columns 37 through 54

26 12 11 13 11 25 17 50 27 56 6 4 6 21 17 9 39 5

Columns 55 through 72

39 16 26 35 28 53 39 24 15 21 27 48 18 56 26 57 22 37

Columns 73 through 90

46 8 43 42 54 57 15 21 18 9 7 24 28 57 43 9 11 25

Columns 91 through 108

40 9 6 41 10 35 19 24 20 14 28 36 43 56 18 50 22 52

Columns 109 through 126

55 21 28 57 44 57 23 51 10 61 44 62 54 25 33 9 45 57

Columns 127 through 144

49 14 6 8 59 5 49 61 27 51 45 52 40 50 46 57 47 24

Columns 145 through 162

59 16 18 9 55 9 43 21 10 25 39 42 6 16 26 13 10 24

Columns 163 through 180

27 61 44 48 15 4 15 5 1 16 44 58 60 9 18 61 6 48

Columns 181 through 198

7 62 44 12 43 24 15 25 60 42 19 61 20 4 44 51 6 5

Columns 199 through 216

40 61 49 14 6 56 45 13 60 16 44 37 55 14 40 42 40 52

Columns 217 through 234

18 25 43 36 55 14 44 52 54 41 26 14 64 53 7 8 27 57

Columns 235 through 250

3 16 11 50 43 48 22 58 44 41 7 53 23 8 49 56

Note: 1) The solution route path begins with a cut at point #31, continues through a final fill at point #56, and then returns to start to complete the Hamiltonian Circuit.

2) Data Points #30, 32, 34, 38 are not on route. The No-Go/Off-Limits area is avoided.

Two sets of trials were performed using the simulated annealing and threshold accepting methods. These two methods were chosen because they had performed better on larger problem instances in previous experiments. The summary of their performance on the Hawaii data is shown in Figure 4.7.

Table 4.7 Hawaii: Table Data for Simulations

Hawaii Illustration

Elevation Factor, *p*, is 250

Run Title	Avg Run Time (min)	Best Cost (min)	Avg Total Cost (min)
PC64-HI-S-4-100-10-10,000	3788	360	368
U64-HI-T-4-100-10-10,000	3490	357	368
PC64-HI-S-4-1000-10-10	38.70	363	367
U64-HI-T-4-1000-10-10	36.12	364	367

The Hawaii illustration confirmed many of the insights that the Egypt example showed. The Unix system remained slightly faster than the PC system. The model found lower cost answers when the number of inner loops was increased. The best single iteration solution was achieved when the number of replications was greatest. This suggests that the best strategy would be to increase both inner loops and replications. Additionally, the processing times were comparable to the previous example for like problem instances.

The analysis of the cost estimate was the most insightful. First of all, the entire Hawaii illustration was only feasible because the slope factors were manually assigned to the Up Hill 3 category. The maximum grade that the type 4 vehicle (785B) can negotiate is 35%. The majority of the point-to-point slopes were over 60%. The initial runs were assigning the MAX slope factor of 1,000,000 minutes and the cost estimates were outrageous. This was because 30% of the route segments were MAX slope and the number of evaluated solutions was such a small % of the total possible solutions. Every evaluated route path included a segment with MAX slope and the time estimates were in the millions of minutes. Trials were performed on runs as large as 10,000,000 iterations, which may seem like a lot. However, it is less than 3.0 x 10^{-484}% of the 250! number of total possible solutions. It made sense that none of the route paths found a way around so many MAX slope areas. Therefore, the first conclusion was that the CF2 model can only accommodate a finite number MAX slope segments. This illustration showed that the maximum number of segments that can be MAX slope is an undetermined number less than 30%. The number of segments is a function of the site dimensions, the elevation factor and the number of iterations. In reasonable processing times, the model could not find a way to navigate around them to level the terrain. Some of the initial runs lasted three days and still returned cost estimates in the millions of minutes. Once all of the MAX slope factors were manually set to Up Hill 3 slope factors, the cost estimates yielded by the model were 367 and 368 minutes.

Another conclusion was that it is virtually impossible to level Hawaii's most dramatic terrain by driving a dump truck around. Typically, explosives initially reshape this type of dramatic terrain, until it can be driven on. This identified a general upper bound to the kinds of problems for which this model is suited.

These estimates were based on the conditions of Up Hill 3 slopes and Very Slow-Go soil categories. These estimates also demonstrated the ability of the CF2 model to produce more accurate estimates than the CF1 model. The CF1 model estimate for leveling this site area was 105 minutes. Remember, this estimate assumes that the slope is zero in all directions, the soil is ideal and that no off-limits areas exist. The CF2 model more clearly represents the rugged, extreme terrain of Hawaii. Budget decisions made on the 105-minute estimate would fail miserably.

So that parallel cost comparisons could be drawn between the Egypt and Hawaii results, the relative elevations were scaled by a factor of ten. The elevation factor for the 64-point Hawaii example summed to $p = 2500$. The terrain is so steep that, on average, a 25-meter movement accompanied a 40-meter change in elevation. Thus, the reason for the manual adjustment of all the MAX slopes to the Up Hill 3 category.

The Hawaii cost estimates were more easily compared to the Egypt estimates when the values for p were close. Egypt's 64-point cost estimate, with $p = 240$, was 192 minutes. The Hawaii 64-point cost estimate, with $p = 250$, was 368 minutes. Since both site areas were 400m x 400m, CF1 model cost estimates only varied slightly. This is because terrain conditions are not considered in CF1 and only the travel times in ideal conditions were compared. However, the physical terrain conditions, of Egypt and

Hawaii, do vary greatly. It takes a lot longer to negotiate the Hawaiian terrain. The differences in the physical terrain conditions are reflected in the avg. total cost estimates.

The final point of analysis involves the impact of scaling the relative elevations by ten, in the Hawaii illustration. The impact was that the 368-minute cost estimate was for a vehicle with 10 times the capacity of a type 4 (785B) dump truck. What was the cost for a single type 4 dump truck to perform the task? For the single 785B truck to do the work of 10, many return trips between cuts and fill were required. The resulting time to negotiate the solution path was 6560 minutes (18 times longer). Considering that an extra roundtrip per move for each additional truck was required to deliver equal payload, the 4.5-day estimate made sense. Travel time alone would require the 785B truck moving non-stop about a 400m x 400m area for 4.5 days. The route path consisted of 1250 cuts and 1250 fills for a total of 2500 loads. Additionally, refueling stops, driver stops, and the pauses for all of the cuts and fills to be performed would further extend the cost estimate.

The Hawaii illustration yielded some useful insights and model limitations. The impacts of area coverage, vehicle capacity and extreme terrain are highlighted. The experiments further demonstrated a need to involve multiple trucks in the leveling process.

CHAPTER 5: CONCLUSIONS

5.1 FINDINGS

The first sets of conclusions are illustrated by the Egypt case study's 64-point data set. The trends in the experiments on smaller problem sizes were confirmed in the trials for the 64-point data set. The information on the model's performance is consolidated in Table 5.1.

Table 5.1 Egypt: Table Data for 64-Point Simulations

Run Title	Run Time (min)	Best Cost (min)	Avg Cost (Std Dev)
PC64-EG-S-3-100-10-10,000	3740.23 (2.5 days)	169 (at rep # 7,012)	192
U64-EG-T-3-100-10-10,000	3442.68 (2.5 days)	170 (at rep # 9035)	192
PC64-EG-S-3-100-10-1000	374.48 (6 hrs)	182	194
U64-EG-T-3-100-10-1000	343.59 (6 hrs)	182	194
PC64-EG-S-3-100-10-10	3.42	180	197
U64-EG-T-3-100-10-10	3.52	180	197
PC64-EG-S-3-1000-10-10	37.29	178	192
U64-EG-T-3-1000-10-10	34.86	178	192

Table 5.1 illustrates some key conclusions. First, problem sizes larger than 100,000 iterations will consistently be solved faster on the Unix *Sunblade 1000* vs the Dell P3 PC. The efficiencies gained by the Unix system for these trial sizes decreases the average solution time by 20%. The reasons for these gains were discussed in section 4.2 and 4.4. Secondly, performing more iterations (inner x outer x reps) leads to better results. However, the # of inner loops is the driving factor. Perform at least 10 replications in a trial to ensure consistent averages. Begin by increasing the number of inner loops (especially if the time until the answer is needed is short). The inner loops are the core of the algorithms. Increasing the number of them is like putting longer legs on a runner. As inner loops increased, the local search and Monte Carlo search algorithms did not

perform as well as the simulated annealing and threshold accepting techniques. This was due to the more sophisticated search strategies of the two latter methods. As long as the inner loops were small, the playing field was level and each technique was equally likely to find a minimum cost. The larger inner loops gave the neighborhood and cooling strategies, of the simulated annealing and threshold accepting methods, a longer opportunity to search before having to repeat the inner loop from a new location. Consequently, the trials with the larger inner loops yielded the lower cost estimates. For example, the 1000 inner loop trial, in the table above, performed 10 replications and found an average solution of 192 in 37 minutes. The same trial with only 100 inner loops ran for 2.5 days and performed no better after completing 10,000 replications.

This is not to imply that replications should be fixed at 10 for all trials. You can see from the 64-point Egypt example that the best cost for any iteration was 169 minutes. This occurred in the 10,000-replication trial. The best time for the 10-replication trial was 178 minutes. Clearly, when you repeat the inner/outer loop sequence 1000 times more often, then the likelihood increases of beginning the shorter inner loop search in a location that is near enough to find a very low minimum. The figure in Appendix D shows how a better solution may eventually occur if enough replications are repeated.
A 171 minute cost estimate was found in replication #1210. It remained the overall best solution until replication #9035 found a solution route path requiring only 170 minutes. However, this is too much like guessing and not dependable for a long run strategy. Therefore, increasing the number of inner loops is a better strategy for finding lower average cost estimates.

5.2 GENERAL CONCLUSIONS

This research verified that some programming routines were not size dependent. Doubling the size of the problem did not double the processing time. It only increased it by approximately 80%. This was a consistent result throughout the experiments.

The research yielded some interesting insights regarding the parameters that define the system. For problems involving terrain with similar dimensions, increasing the number of evaluated data points provides more dense coverage to the site area. Dense coverage of an area improves resolution and allows for a smoother leveled appearance.

However, the increase in data points also increases the size of the solution space, thus addressing the problem requires more computational effort. On the other hand, fewer data points have a smaller solution space, but will usually leave terrain appearing lumpy.

The elevation factor, *p*, the site dimensions and the number of data points work hand-in-hand to define the problem size. The total number of cuts and fills dictates the number of loads you relocate. The data points dictate where you relocate them. The site dimensions dictate how long you travel to relocate them. For this reason, a condensed site with dramatic contours could take as long or longer than a smoother site with much larger dimensions. For example, leveling a 100m x 100m section of the Hawaii illustration cost more than leveling the 400m x 400m site in the Egypt scenario.

Another result was that an upper limit exists for the effects of soil and slope on the model's answers. This limit applies to the vehicles studied in this research and any vehicles whose technical specifications are comparable. The limit implies that the worst possible, yet navigable, conditions exist throughout the site area. It is a worst case estimate and is approximately equal to 4 times the travel time estimate in ideal conditions. Let **X** = travel time estimate in ideal conditions. For problems of equal values for *p* and for the same number of data points, let **Z** = the travel time required to level the site. The following relationship holds true in the CF2 model.

$$\mathbf{X} \leq \mathbf{Z} \leq 4\mathbf{X}$$

This is calculated using the adjustment factors for slope and soil for each particular vehicle. This upper bound is independent of terrain. As long as *p* and the number of points are consistent, then the upper bound for the time required for leveling the terrain will always be true for the CF2 model as it is currently designed.

The model is sensitive to the elevation chosen as the field level elevation. The impact increases as the field level elevation varies from the "median" site elevation. Consider a problem instance where the elevation range was 2050 to 2070 feet, with field level at the median elevation of 2060 feet. If the desired elevation were 2051 feet, route times would dramatically increase. Routes would require many more trips to an "off-site" dumping location to relocate the excess dirt.

Now the ability exists to get reasonably effective results for moderately sized problems. Problem instances most useful for the land leveling applications usually have a *p* = 100 or more. Egypt is far from rugged, dramatic terrain, yet its 64-point scenario had a p = 560 for a 400 x 400m area. The smart neighborhood strategies and search techniques coupled with superior computer speed produced some useful results in a timely manner.

Regarding how many data points to choose, the conclusion is that it depends upon the size of the terrain, the elevation factor and the capacity of the earthmoving vehicle. The experiments on the 64-point data set were illustrated graphically using 64 points to illustrate the terrain area. Unless every other point in the site area was at field level before the process began, then using more initial points would have caused the 64-point diagram to not be so flat. On the physical terrain, the 20m x 20m x 1 foot sections that comprised each load for the 777D truck represented an average of 4 points per 100m x 100m area. 25 points per 100m x 100m provides 100% coverage to a site area of this size. The conclusion was that either a 400-point data set or a 6.25 times larger capacity vehicle was needed to fully cover an area of this size.

The values for 100% coverage are upper bounds values, because some of the points may already be at field level. In this case, fewer points or a smaller capacity vehicle could be used. The conclusion is that individual dump truck applications, for the size vehicles discussed in this research, are best suited for terrain up to 200m x 200m in size. This conclusion is based on the capacities of standard commercial dump trucks and the data point limit for the Mustard Seed Data Builder. The baseline model for these coverage estimates was the 777D truck. Examples of coverage for this truck are listed below. For smaller vehicles, such as the 775D Quarry truck, the coverage was as low as 44% using 64 data points. For larger vehicles, such as the 785B truck, the coverage was 100% using 64 data points.

For the 777D Caterpillar Dump Truck:

Linear Cut / Fill Dimensions (1-foot depth)	**# of Data Points Needed (for 100% area coverage)**
50m x 50m	6.25
100m x 100m	25
200m x 200m	100
400m x 400m	400

Table 5.2 Area Coverage Table for 777D Caterpillar Dump Truck

Area coverage is a tool that ensures you consider the entire site. The resolution to which you level the land increases as the number of data point increases until the point at which cut and fill locations overlap. For example, 25 data points provides 100% coverage to a 100m x 100m area for a 777D truck (see Table 5.2). Additional points would cause some dirt to be relocated twice, because the cut and fill areas would overlap. It is not important to always achieve 100% coverage. The important thing is to level the points of interest to the construction planner. The CF2 model is flexible because it does not rely on a unit square grid and allows planners to handpick the points of interest to them. By visual inspection of the terrain or a map, it is easy to determine where dirt needs to be relocated. Consider the 25 data point example in the figure above. If 20 data points cover 80% of the area and the other 20% is 1 foot away from field level, then it may be preferred to only use 20 data points. Considering that 5 areas are close to field level, then the time saved from the truck's 5 fewer stops may offset the slightly less accurate 20-point estimate provided by the CF2 model. The needs of the user should always dictate decisions for how the CF2 model is implemented.

The accuracy in the answer, like most results, ultimately depends on money—How soon is the answer needed? 2 months may be fine for next year's bid on a highway construction project, but 2 minutes may be too long to locate the downed pilot with the unmanned aerial surveillance drone. How exact does the answer need to be? Ten dollars saved might be negligible to a $10,000 project bid, but ten minutes saved could be the margin of victory for the winner of the race for buried treasure.

5.3 FUTURE DIRECTIONS

The technology is fast approaching that will eliminate the need for human drivers in dump trucks and other construction equipment. Consider the efficiencies gained by reshaping terrain by pinpointing exactly where earth needs to be relocated and only moving between those locations to accomplish the job. This research supports these extensions in the state of the art in the construction industry.

This is an innovative approach to an intractable discrete optimization problem and represents another step in the process of optimizing vehicle routing operations using state of the art solution techniques. Although not exhaustive, this model is robust in that it accounts for what I feel are the major aspects of the system. This research demonstrated that the CF2 model effectively represents the critical elements of the land-leveling process. The abstraction of measurable elements was appropriate and the algorithms smoothly integrated them. Hauling dirt isn't rocket science; therefore the process is not overly complicated. An earth-moving vehicle is most affected by the ground surface, the slope along each move and the presence of off-limit hazards along the way. CF2 appropriately accommodated these attributes. However, more research is needed to advance the progress made in this research.

There are two major aspects of land leveling that this model does not address--Dynamic Terrain and Fleet Optimization. Future evolutions of the model could recalculate point-to-point slopes as each cut is taken and each fill is deposited. The worst case cost estimates would improve if slopes were dynamic in the CF2 model. Because the slopes are static in the CF2 model, the estimates that it generates assumes the slope for successive visits to a location is along the same as the initial level of grade. The impact of static slope on the accuracy of the estimates increased with the value for the elevation factor, p. In the Egypt case study, the impact was slight because the slope range was small and the distances were short. In the Hawaii example, the impact was great because the slope range was large and the distances were longer. The effect of this in the model caused the truck to travel steeper paths for longer distances. Dynamic slope calculations would reflect the change in elevations as they converge on field level. The resulting cost estimates would be more accurate.

Expanding the soil and slope categories is another useful idea. CF2 restricts all the earth surfaces into four categories. CF2 groups the slopes into increments of 10 or 15. The addition of more categories would allow for more specific assignments and refine the accuracy of the cost estimates.

Because of the lack of coverage that a single dump truck can offer, another conclusion was to consider the use of multiple dump trucks. The idea of fleet operations is a concept for future research that will make the model more useful in larger problem sizes. The model could also address ways to identify optimal combinations of vehicles, with their own capabilities, that may do the same job cheaper than a single dump truck.

Traditional methods of vehicle route optimization represent haul roads as a network with impedances on each network link. This approach has the potential to identify ideal haul road locations from site topography. This approach reduces reliance on expert systems and expert opinion regarding how to route earthmoving vehicles along terrain while leveling the terrain. This approach also brings land leveling operations one step closer to robotic, driver-free construction operations.

The techniques evaluated in this research performed well. However, new search techniques should be explored as they become appropriate. The four methods in this research are a very good sampling of available techniques, but not exhaustive. In 1996, a meta-heuristic was developed that combines the strategies of simulated annealing and local search by embedding the local search technique into the simulated annealing algorithm. This method and other types of innovative and creative approaches might prove useful to explore and compare to the results in this research.

Additionally, one can easily appreciate the requirement for high power computational aids and understand why limited research has occurred in this specific area when you realize how quickly the instances grow in size (see Table 5.3).

Table 5.3 Illustration of Problem Size Growth

p (# of loads)	*(n-1) possible solution routes*
4	3! = 6
5	4! = 24
6	5! = 120
…	…
10	9! = 362,880
…	…
16	15! = 1,307,674,368,000

For example, a 16-point data set, where all the coordinates are either +1 (cut) or –1 (fill), is a problem instance with p = 16. To enumerate and solve this problem, you see that over 1 trillion routes must be evaluated. For older systems that performed thousands of operations per second, the solution time is 20.73 days. For the *SunBlade 1000* that performs millions of operations per second the solution time is 7.57 days, much faster. Nevertheless, it takes a week to iteratively solve a small instance. Since optimal solutions are not attainable for problems sufficiently large enough to be useful, future researchers could also examine estimate accuracy. An interesting point of inquiry would be determining the degree of accuracy for the cost estimates, using confidence intervals, for certain solution techniques, or number of iterations. This could determine the relative improvements in the estimates as a greater percentage of the total solutions are evaluated. This research advanced the Cut-Fill Model one step closer to being an effective tool in construction and other vehicle routing applications. The CF2 model provides useful, conservative, near-optimal estimates. The use of the CF2 model should be based on visual inspection of the terrain or a map, the user's needs, and the answers to the questions:

- How accurate does the cost estimate need to be?
- To what degree of detail does the leveling need to be?
- How long until the answer is needed?

The CF2 model greatly benefits the construction industry, and its corresponding solution techniques have great potential to make advances in the other applications mentioned in this research. This work on the CF2 model and future work in this area will advance the likelihood of operations by fleets of automated earthmoving vehicles, without human drivers, that use global positioning and geographic information systems to maneuver in the future. What a great day that will be, unless, of course, you drive a dump truck for a living!

APPENDIX A

PC Verification Runs (Sample Data)

%-- 1/23/02 --%
dougshovel3x3
10
10
10
dougshovel7x7
10
10
10
%-- 1/25/02 --%
dougshovel3x3simul
3
3
3
dougshovel3x3simul
3
3
3
dougshovel3x3monte
3
3
3
dougshovel3x3monte
5
5
5
dougshovel3x3local
3
3
3
dougshovel3x3test
3
3
3
dougshovel3x3test
3
3
3

%-- 1/28/02 --%
dougshovel3x3local
3
3
3
dougshovel3x3test
3
3
1
dougshovel3x3test
5
5
10
dougshovel3x3test
5
5
3
dougshovel3x3simul
5
5
5
dougshovel3x3simul
5
5
5
dougshovel3x3simul
6
6
6
%-- 1/28/02 --%
dougshovel7x7
10
10
10
%-- 1/29/02 --%
dougshovel7x7
100
100
1
dougshovel7x7
10
10
5
dougshovel7x7
10
10
10

%-- 1/30/02 --%
dougshovel3x3test
5
5
5
dougshovel3x3test
10
10
10
dougshovel3x3
10
10
10
dougshovel7x7
20
20
20
dougshovel7x7
15
15
15
dougshovel7x7
10
10
10
dougshovel7x7
100
100
100
%-- 2/05/02 --%
dougshovel3x3test
25
25
25
dougshovel7x7
25
25
50
dougshovel7x7
50
10
10
dougshovel3x3test
10
25
10

APPENDIX A

PC Verification Runs (Sample Data)

dougshovel7x7
10
10
10
dougshovel7x7
10
10
10
dougshovel7x7
5
5
5
dougshovel7x7
5
5
5
dougshovel7x7
5
5
5
dougshovel7x7
10
10
10
dougshovel7x7
10
10
10
dougshovel3x3
10
10
10
dougshovel3x3
10
10
10
%-- 2/16/02 --%
dougshovel3x3
10
10
10
dougshovel3x3
10
5
5

%-- 2/18/02 --%
dougshovel3x3
3
3
3
dougshovel3x3monte
3
3
3
dougshovel3x3simul
5
5
5
dougshovel7x7
5
5
5
dougshovel7x7
5
5
5
dougshovel7x7
5
5
5
dougshovel7x7
3
3
3
%-- 2/23/02 --%
dougshovel3x3local
3
3
3
dougshovel3x3local
3
3
3
dougshovel3x3monte
5
5
5
dougshovel3x3monte
5
5
5
dougshovel3x3simul
5
5
5

%-- 2/24/02 --%
dougshovel3x3
1
3
3
%-- 2/24/02 --%
dougshovel3x3
1
1
1
dougshovel3x3
1
1
1
dougshovel3x3
1
1
1
dougshovel3x3
5
5
5
dougshovel3x3
2
2
2
dougshovel3x3
5
5
5
dougshovel3x3
10
10
10
dougshovel3x3
5
5
5
%-- 2/25/02 --%
dougshovel7x7
10
10
10

Unix Verification Runs (Sample Data)

%-- 2/08/02 --%	%-- 2/08/02 --%
DOUGSHOVEL TA	***DOUGSHOVEL simul***
25	25
25	25
25	25
DOUGSHOVEL monte	***DOUGSHOVEL local***
50	50
50	50
50	50
%-- 2/11/02 --%	%-- 2/11/02 --%
DOUGSHOVEL TA	***DOUGSHOVEL simul***
15	15
15	15
5	5
DOUGSHOVEL local	***DOUGSHOVEL monte***
5	***5***
10	10
10	10
DOUGSHOVEL simul	***DOUGSHOVEL TA***
5	5
15	15
10	10
DOUGSHOVEL monte	***DOUGSHOVEL local***
5	5
10	10
15	15
DOUGSHOVEL TA	***DOUGSHOVEL simul***
5	5
5	5
5	5
DOUGSHOVEL monte	***DOUGSHOVEL local***
5	5
5	5
5	5

PC Verification Runs (Real Data)

%-- 3/02/02 --%	%-- 3/03/02 --%
Thesiscode9x9 simul	***Thesiscode7x7 TA***
2	50
2	50
2	50
Thesiscode7x7 TA	***Thesiscode7x7 simul***
5	50
5	50
5	50
Thesiscode7x7 local	***Thesiscode7x7 local***
10	50
20	50
20	50
Thesiscode7x7 monte	***Thesiscode7x7 monte***
10	50
50	50
50	50
	Thesiscode7x7 TA
	20
	20
	20
	Thesiscode7x7 simul
	20
	20
	20

APPENDIX A

Unix Verification Runs (Real Data)

%-- 3/04/02 --%
Thesiscode simul
3
3
3
Thesiscode monte
2
2
2
Thesiscode local
2
2
2
Thesiscode TA
2
2
2
Thesiscode simul
10
10
100
Thesiscode simul
10
10
500
Thesiscode simul
10
10
500
%-- 3/05/02 --%
Thesiscode simul
10
10
2000

%-- 3/06/02 --%
Thesiscode TA
20
20
10
%-- 3/06/02 --%
Thesiscode TA
2
2
5
Thesiscode TA
2
2
2
Thesiscode TA
2
2
2
Thesiscode monte
2
2
2
Thesiscode local
2
2
2

APPENDIX B: CODE FOR THE (SRCFP) ALGORITHM

```
clear;
start=cputime;
M=input('How Many Inner Loop Iterations?   ');
K=input('How Many Outer Loop Iterations?   ');
Reps=input('How Many Replications?   ');

Distancematrix=wk1read('filename.wk1');
initialvector=wk1read('filename.wk1');

%Start of Replication
for n=1:1:Reps;
  n                                              % counter
  atsolution=newroute2(solution1new(Distancematrix, initialvector));

  for i=1:1:(length(atsolution))/2               %permutes the initial solution
   minus(i)=atsolution(2*i);
   plus(i)=atsolution(2*i-1);
  end

  minusnew=perm(minus,(length(atsolution))/2);
  plusnew=perm(plus, (length(atsolution))/2);

  for i=1:1:(length(atsolution))/2
    atsolution(2*i)=minusnew(i);
    atsolution(2*i-1)=plusnew(i);
  end

    w=cost1(atsolution,Distancematrix);

  bestcosttodate=w;
  bestsolutiontodate=atsolution;
  j=1;

  %beta=.85;
  %Tk = 2335;                          %initial temperature

 %Start of Outer Loop
   for k=1:1:K

   %R(k)=0;                            %R(k) for Deterministic Local Search  (Method 1)
   %R(k)=1000000;                      %R(k) for Monte Carlo Search          (Method 2)
   %Rk= -Tk*log(rand);                 %R(k) for Simulated Annealing         (Method 3)
   %Rk=Tk*beta;                        %R(k) for Threshold Accepting         (Method 4)

   %Tk=Tk*beta;                         %cooling schedule

 %Start of Inner Loop
    for m=1:1:M
     neigh= newroute2(atsolution);      %finds neighbor, does edge switches
     w = cost1(neigh,Distancematrix);
     delta=w-cost1(atsolution,Distancematrix);
     if delta<Rk
```

```
    atsolution=neigh;
         if w < bestcosttodate
            bestcosttodate=w;
            bestsolutiontodate=atsolution;
         end
      end
      iterationbestcosttodate(n,j)=bestcosttodate;
      j=j+1;
      m=m+1;
    end
  %End of Inner Loop

    k=k+1;

   end
  %End of Outer Loop

   bestcostforreplication(n,:)=bestcosttodate;
 % n=n+1
   bestcosttodate
end
%End of Replication

bestcostforreplication
averagecost=mean(bestcostforreplication)
standarddev=std(bestcostforreplication)
bestcost=min(bestcostforreplication)
worstcost=max(bestcostforreplication)

for i=1:1:length(iterationbestcosttodate(1,:))
    individualcost(i)=mean(iterationbestcosttodate(:,i));
    standarddevindividual(i)=std(iterationbestcosttodate(:,i));
end

for q=1:length(bestcostforreplication);
    averagecostline(q)=averagecost;
end
plot(bestcostforreplication,'b')
hold
plot(averagecostline,'r')

figure
plot(individualcost,'b')
hold
plot(individualcost-standarddevindividual,'r')
plot(individualcost+standarddevindividual,'r')

finish=cputime;
time=finish-start
```

SUB-ROUTINES INTERNAL TO THE CUT-FILL CODE

NEW ROUTE 2 CODE

```
%This finds a neighbor doing edge switches

function x=newroute2(route)
r1=0;
r2=1;
while mod(abs(r1-r2),2)~=0 | r1==r2
r1=floor(rand*length(route)+1);
r2=floor(rand*length(route)+1);
end


 if r1<=r2
z=route(r1:r2);
z2=fliplr(z);
x=[route(1:r1-1),z2,route(r2+1:length(route))];
else
   z=[route(r1:length(route)),route(1:r2)];
   z2=fliplr(z);
   x=[z2(length(route)-r1+2:length(z2)),route(r2+1:r1-1),z2(1:length(route)-r1+1)];
end
```

PERM

```
%This permutes the locations of the cuts and the fills

function x = perm (y,length)
a=randperm(length);
       for i=1:1:length
       x(a(i))=y(i);
    end
```

COST 1

```
%Given a route this computes the distance

function x=cost1(route,Distancematrix)
for j=2:1:length(route)
      d(j)=Distancematrix(route(j),route(j-1));
         end
     x=sum(d)+Distancematrix(route(length(route)),route(1));
```

VARIABLES USED

Name

Distancematrix
K
M
Reps
Rk
Tk
atsolution
averagecost
averagecostline
bestcost
bestcostforreplication
bestcosttodate
bestsolutiontodate
beta
delta
finish
i
individualcost
initialvector
iterationbestcosttodate
j
k
m
minus
minusnew
n
neigh
plus
plusnew
q
standarddev
standarddevindividual
start
time
w
worstcost

APPENDIX C

DATA SET:

DATA16EG-3-140

Points 1-16

VEHICLE	#
773D	*1*
775 Quarry	*2*
777D	*3*
785B	*4*
Scraper	*5*

VEHICLE # 3

Speed Data

loaded speed	20	mph
empty speed	38	mph

Slope Data

Max Up	35	1000000
Up Hill 3	25	1.8
Up Hill 2	15	1.5
Up Hill 1	5	1.2
Flat	-5	1
Down Hill 1	-15	0.9
Down Hill 2	-25	0.8
Down Hill 3	-35	0.7
Max Down	-100	1000000

Soil Data

Go Soil	1
Slow-Go Soil	1.3
Very Slow Soil	2
Off-Limit/No-Go	1000000

	x-coord	*y-coord*	*elevation*	*soil type*
Point #1	2565	9055	140	3
Point #2	2560	9060	125	3
Point #3	2552	9048	125	3
Point #4	2555	9040	125	3
Point #5	2575	9055	125	3
Point #6	2562	9070	125	3
Point #7	2550	9065	125	3
Point #8	2530	9070	125	3
Point #9	2520	9060	125	3
Point #10	2530	9055	125	3
Point #11	2540	9063	110	2
Point #12	2545	9063	110	2
Point #13	2542	9058	110	2
Point #14	2540	9045	100	2
Point #15	2545	9045	100	2
Point #16	2515	9015	125	2
Point #17	0000	0000	0	0
Point #18	0000	0000	0	0
Point #19	0000	0000	0	0
Point #20	0000	0000	0	0
Point #21	0000	0000	0	0
Point #22	0000	0000	0	0
Point #23	0000	0000	0	0
Point #24	0000	0000	0	0
Point #25	0000	0000	0	0
Point #26	0000	0000	0	0
Point #27	0000	0000	0	0
Point #28	0000	0000	0	0
Point #29	0000	0000	0	0
Point #30	0000	0000	0	0
Point #31	0000	0000	0	0
Point #32	0000	0000	0	0

Howard D. McInvale
ISE Spring 2002

	Point #1	Point #2	Point #3	Point #4	Point #5	Point #6	Point #7	Point #8	Point #9	Point #10	Point #11	Point #12	Point #13	Point #14	Point #15	Point #16
Point #1	0.00	0.22	0.38	0.46	0.26	0.39	0.46	0.98	1.16	0.90	0.67	0.55	0.60	0.69	0.69	1.65
Point #2	0.15	0.00	0.37	0.53	0.41	0.26	0.29	0.81	1.03	0.78	0.52	0.39	0.47	0.64	0.55	1.64
Point #3	0.38	0.37	0.00	0.22	0.62	0.62	0.44	0.80	0.88	0.59	0.49	0.43	0.36	0.38	0.23	1.27
Point #4	0.46	0.53	0.22	0.00	0.64	0.79	0.66	1.00	1.04	0.75	0.71	0.64	0.57	0.41	0.34	1.21
Point #5	0.26	0.41	0.62	0.64	0.00	0.51	0.69	1.22	1.42	1.16	0.92	0.80	0.85	0.94	0.81	1.85
Point #6	0.39	0.26	0.62	0.79	0.51	0.00	0.33	0.82	1.11	0.91	0.59	0.47	0.60	0.86	0.78	1.86
Point #7	0.46	0.29	0.44	0.66	0.69	0.33	0.00	0.53	0.78	0.57	0.26	0.17	0.27	0.57	0.53	1.57
Point #8	0.98	0.81	0.80	1.00	1.22	0.82	0.53	0.00	0.36	0.39	0.31	0.43	0.44	0.69	0.75	1.46
Point #9	1.16	1.03	0.88	1.04	1.42	1.11	0.78	0.36	0.00	0.29	0.52	0.65	0.57	0.64	0.75	1.16
Point #10	0.90	0.78	0.59	0.75	1.16	0.91	0.57	0.39	0.29	0.00	0.33	0.44	0.32	0.44	0.46	1.10
Point #11	0.51	0.39	0.37	0.53	0.69	0.44	0.20	0.24	0.39	0.25	0.00	0.10	0.10	0.35	0.36	1.04
Point #12	0.42	0.29	0.32	0.48	0.60	0.35	0.08	0.32	0.49	0.33	0.10	0.00	0.11	0.36	0.35	1.09
Point #13	0.45	0.35	0.27	0.43	0.64	0.45	0.20	0.33	0.43	0.24	0.10	0.11	0.00	0.25	0.26	0.98
Point #14	0.52	0.48	0.19	0.30	0.70	0.64	0.43	0.52	0.48	0.22	0.35	0.36	0.25	0.00	0.10	0.75
Point #15	0.34	0.41	0.12	0.17	0.61	0.58	0.40	0.56	0.56	0.35	0.36	0.35	0.26	0.10	0.00	0.82
Point #16	1.23	1.64	0.96	0.91	1.39	1.39	1.18	1.10	0.87	0.82	1.04	1.09	0.98	0.75	0.82	0.00

Data16EG-3-140
Integrated Time Matrix (1 of 2)

Pts: 16
Location: Egypt vic Grid 90..25..
Truck Type: 3 (777D Caterpillar Dump)
Elevation Factor: 140 (Total cuts and fills)

+20	+5	+5	+5	+5	+5	+5	+5	+5	+5	-10	-10	-10	-20	-20	+5

Data16EG-3-140a
Initial Vector of Relative Elevations (1 of 2)

Pts: 16
Location: Egypt vic Grid 90..25..
Truck Type: 3 (777D Caterpillar Dump)
Elevation Factor: 140 (Total cuts and fills)

APPENDIX C

DATA SET:
DATA16EG-3-140
Points 17-32

VEHICLE	#
773D	*1*
775 Quarry	*2*
777D	*3*
785B	*4*
Scraper	*5*

VEHICLE # 3

Speed Data

loaded speed	20	mph
empty speed	38	mph

Slope Data

Max Up	35	1000000
Up Hill 3	25	1.8
Up Hill 2	15	1.5
Up Hill 1	5	1.2
Flat	-5	1
Down Hill 1	-15	0.9
Down Hill 2	-25	0.8
Down Hill 3	-35	0.7
Max Down	-100	1000000

Soil Data

Go Soil	1
Slow-Go Soil	1.3
Very Slow Soil	2
Off-Limit/No-Go	1000000

	x-coord	*y-coord*	*elevation*	*soil type*
Point #2	0000	0000	0	0
Point #3	0000	0000	0	0
Point #4	0000	0000	0	0
Point #5	0000	0000	0	0
Point #6	0000	0000	0	0
Point #7	0000	0000	0	0
Point #8	0000	0000	0	0
Point #9	0000	0000	0	0
Point #10	0000	0000	0	0
Point #11	0000	0000	0	0
Point #12	0000	0000	0	0
Point #13	0000	0000	0	0
Point #14	0000	0000	0	0
Point #15	0000	0000	0	0
Point #16	0000	0000	0	0
Point #17	2525	9065	110	3
Point #18	2525	9055	125	3
Point #19	2525	9045	125	3
Point #20	2525	9040	110	3
Point #21	2545	9075	110	2
Point #22	2545	9065	110	2
Point #23	2542	9050	110	2
Point #24	2565	9058	130	3
Point #25	2550	9070	125	3
Point #26	2550	9055	110	3
Point #27	2550	9045	110	3
Point #28	2565	9052	130	3
Point #29	2555	9065	125	3
Point #30	2565	9060	125	3
Point #31	2562	9053	130	3
Point #32	2516	9016	125	3

Howard D. McInvale
ISE Spring 2002

APPENDIX C

	Point #17	Point #18	Point #19	Point #20	Point #21	Point #22	Point #23	Point #24	Point #25	Point #26	Point #27	Point #28	Point #29	Point #30	Point #31	Point #32
Point #17	0.00	0.26	0.51	0.64	0.57	0.51	0.58	1.04	0.66	0.69	0.82	1.08	0.77	1.04	1.00	1.28
Point #18	0.26	0.00	0.26	0.39	0.73	0.57	0.46	1.03	0.75	0.64	0.69	1.03	0.81	1.04	0.95	1.03
Point #19	0.51	0.26	0.00	0.15	0.93	0.73	0.46	1.08	0.91	0.69	0.64	1.04	0.93	1.10	0.97	0.78
Point #20	0.64	0.39	0.10	0.00	1.04	0.82	0.51	1.13	1.00	0.75	0.66	1.07	1.00	1.15	1.01	0.66
Point #21	0.43	0.55	0.69	0.78	0.00	0.19	0.49	0.51	0.11	0.40	0.59	0.59	0.27	0.48	0.54	1.27
Point #22	0.39	0.43	0.55	0.62	0.19	0.00	0.29	0.41	0.11	0.22	0.40	0.46	0.19	0.40	0.40	1.10
Point #23	0.44	0.34	0.34	0.38	0.49	0.29	0.00	0.47	0.42	0.18	0.18	0.44	0.38	0.48	0.39	0.82
Point #24	1.04	1.03	1.08	1.13	0.67	0.54	0.63	0.00	0.49	0.39	0.51	0.15	0.31	0.06	0.15	1.66
Point #25	0.66	0.75	0.91	1.00	0.22	0.22	0.55	0.49	0.00	0.39	0.64	0.60	0.18	0.46	0.53	1.64
Point #26	0.69	0.64	0.69	0.75	0.53	0.29	0.24	0.39	0.39	0.00	0.26	0.39	0.29	0.41	0.25	1.33
Point #27	0.82	0.69	0.64	0.66	0.78	0.53	0.24	0.51	0.64	0.26	0.00	0.43	0.53	0.55	0.37	1.15
Point #28	1.08	1.03	1.04	1.07	0.78	0.61	0.59	0.15	0.60	0.39	0.43	0.00	0.42	0.21	0.08	1.56
Point #29	0.77	0.81	0.93	1.00	0.36	0.26	0.51	0.31	0.18	0.29	0.53	0.42	0.00	0.29	0.36	1.61
Point #30	1.04	1.04	1.10	1.15	0.64	0.53	0.64	0.04	0.46	0.41	0.55	0.21	0.29	0.00	0.20	1.69
Point #31	1.00	0.95	0.97	1.01	0.71	0.53	0.52	0.15	0.53	0.38	0.37	0.08	0.36	0.20	0.00	1.52
Point #32	1.28	1.03	0.78	0.66	1.69	1.46	1.10	1.66	1.64	1.33	1.15	1.56	1.61	1.69	1.52	0.00

Data16EG-3-140

Integrated Time Matrix

Pts: 16

Location: Egypt vic Grid 90..25..

Truck Type: 3 (777D Caterpillar Dump)

Elevation Factor: 140 (Total cuts and fills)

-10	+5	+5	-10	-10	-10	-10	+10	+5	-10	-10	+10	+5	+5	+10	+15

Data16EG-3-140a

Initial Vector of Relative Elevations

Pts: 16

Location: Egypt vic Grid 90..25..

Truck Type: 3 (777D Caterpillar Dump)

Elevation Factor: 140 (Total cuts and fills)

Howard D. McInvale
ISE Spring 2002

APPENDIX C

DATA SET:

DATA32EG-3-280

Points 1-32

VEHICLE	#
773D	*1*
775 Quarry	*2*
777D	*3*
785B	*4*
Scraper	*5*

VEHICLE # 3

Speed Data

loaded speed	20	mph
empty speed	38	mph

Slope Data

Max Up	35	1000000
Up Hill 3	25	1.8
Up Hill 2	15	1.5
Up Hill 1	5	1.2
Flat	-5	1
Down Hill 1	-15	0.9
Down Hill 2	-25	0.8
Down Hill 3	-35	0.7
Max Down	-100	1000000

Soil Data

Go Soil	1
Slow-Go Soil	1.3
Very Slow Soil	2
Off-Limit/No-Go	1000000

	x-coord	*y-coord*	*elevation*	*soil type*
Point #1	2565	9055	140	3
Point #2	2560	9060	125	3
Point #3	2552	9048	125	3
Point #4	2555	9040	125	3
Point #5	2575	9055	125	3
Point #6	2562	9070	125	3
Point #7	2550	9065	125	3
Point #8	2530	9070	125	3
Point #9	2520	9060	125	3
Point #10	2530	9055	125	3
Point #11	2540	9063	110	2
Point #12	2545	9063	110	2
Point #13	2542	9058	110	2
Point #14	2540	9045	100	2
Point #15	2545	9045	100	2
Point #16	2515	9015	125	2
Point #17	2525	9065	110	3
Point #18	2525	9055	125	3
Point #19	2525	9045	125	3
Point #20	2525	9040	110	3
Point #21	2545	9075	110	2
Point #22	2545	9065	110	2
Point #23	2542	9050	110	2
Point #24	2565	9058	130	3
Point #25	2550	9070	125	3
Point #26	2550	9055	110	3
Point #27	2550	9045	110	3
Point #28	2565	9052	130	3
Point #29	2555	9065	125	3
Point #30	2565	9060	125	3
Point #31	2562	9053	130	3
Point #32	2516	9016	125	3

Howard D. McInvale
ISE Spring 2002

APPENDIX C

	Point #1	Point #2	Point #3	Point #4	Point #5	Point #6	Point #7	Point #8	Point #9	Point #10	Point #11	Point #12	Point #13	Point #14	Point #15	Point #16	Point #17	Point #18	Point #19	Point #20	Point #21	Point #22	Point #23	Point #24	Point #25	Point #26	Point #27	Point #28	Point #29	Point #30	Point #31	Point #32
Point #1	0.00	0.22	0.38	0.46	0.26	0.39	0.46	0.98	1.16	0.90	0.67	0.55	0.60	0.69	0.69	1.65	1.06	1.03	1.06	1.10	0.73	0.57	0.60	0.09	0.55	0.46	0.56	0.09	0.36	0.15	0.11	1.61
Point #2	0.16	0.00	0.37	0.53	0.41	0.26	0.29	0.81	1.03	0.78	0.52	0.39	0.47	0.64	0.55	1.64	0.91	0.91	0.98	1.04	0.55	0.41	0.53	0.14	0.36	0.29	0.46	0.24	0.18	0.13	0.19	1.60
Point #3	0.38	0.37	0.00	0.22	0.62	0.62	0.44	0.80	0.88	0.59	0.49	0.43	0.36	0.38	0.23	1.27	0.82	0.72	0.70	0.72	0.72	0.47	0.26	0.42	0.57	0.22	0.11	0.35	0.44	0.45	0.29	1.24
Point #4	0.46	0.53	0.22	0.00	0.64	0.79	0.66	1.00	1.04	0.75	0.71	0.64	0.57	0.41	0.34	1.21	1.00	0.86	0.78	0.77	0.94	0.69	0.42	0.53	0.78	0.41	0.22	0.40	0.64	0.57	0.38	1.18
Point #5	0.26	0.41	0.62	0.64	0.00	0.51	0.69	1.22	1.42	1.16	0.92	0.80	0.85	0.94	0.81	1.85	1.31	1.28	1.31	1.34	0.93	0.81	0.86	0.27	0.75	0.64	0.69	0.27	0.57	0.29	0.34	1.82
Point #6	0.39	0.26	0.62	0.79	0.51	0.00	0.33	0.82	1.11	0.91	0.59	0.47	0.60	0.86	0.78	1.86	0.96	1.03	1.15	1.22	0.46	0.46	0.73	0.32	0.31	0.49	0.71	0.47	0.22	0.27	0.44	1.82
Point #7	0.46	0.29	0.44	0.66	0.69	0.33	0.00	0.53	0.78	0.57	0.26	0.17	0.27	0.57	0.53	1.57	0.64	0.69	0.82	0.91	0.29	0.15	0.44	0.43	0.13	0.26	0.51	0.51	0.13	0.41	0.44	1.53
Point #8	0.98	0.81	0.80	1.00	1.22	0.82	0.53	0.00	0.36	0.39	0.31	0.43	0.44	0.69	0.75	1.46	0.22	0.41	0.66	0.78	0.41	0.41	0.60	0.95	0.51	0.64	0.82	1.01	0.66	0.94	0.93	1.43
Point #9	1.16	1.03	0.88	1.04	1.42	1.11	0.78	0.36	0.00	0.29	0.52	0.65	0.57	0.64	0.75	1.16	0.22	0.18	0.41	0.53	0.75	0.66	0.62	1.16	0.81	0.78	0.86	1.17	0.91	1.16	1.09	1.14
Point #10	0.90	0.78	0.59	0.75	1.16	0.91	0.57	0.39	0.29	0.00	0.33	0.44	0.32	0.44	0.46	1.10	0.29	0.13	0.29	0.41	0.64	0.46	0.33	0.90	0.64	0.51	0.57	0.90	0.69	0.91	0.82	1.06
Point #11	0.44	0.34	0.32	0.46	0.60	0.39	0.17	0.20	0.34	0.21	0.00	0.08	0.09	0.30	0.31	0.90	0.25	0.28	0.39	0.46	0.22	0.09	0.22	0.43	0.20	0.21	0.34	0.46	0.25	0.42	0.40	0.88
Point #12	0.36	0.26	0.28	0.42	0.52	0.31	0.08	0.28	0.42	0.28	0.08	0.00	0.10	0.31	0.30	0.95	0.34	0.36	0.45	0.51	0.20	0.03	0.22	0.34	0.13	0.16	0.31	0.38	0.17	0.34	0.33	0.92
Point #13	0.39	0.30	0.24	0.37	0.55	0.39	0.18	0.28	0.37	0.21	0.09	0.10	0.00	0.22	0.22	0.85	0.31	0.29	0.36	0.41	0.29	0.13	0.13	0.38	0.24	0.14	0.25	0.40	0.25	0.39	0.34	0.83
Point #14	0.45	0.42	0.19	0.26	0.61	0.56	0.37	0.45	0.42	0.21	0.30	0.31	0.22	0.00	0.08	0.65	0.42	0.30	0.23	0.26	0.51	0.34	0.08	0.47	0.45	0.24	0.17	0.43	0.42	0.49	0.39	0.63
Point #15	0.34	0.35	0.11	0.17	0.53	0.50	0.34	0.49	0.49	0.30	0.31	0.30	0.22	0.08	0.00	0.71	0.47	0.37	0.33	0.34	0.50	0.33	0.09	0.40	0.43	0.19	0.08	0.35	0.37	0.42	0.31	0.69
Point #16	1.07	1.64	0.83	0.79	1.20	1.21	1.02	0.95	0.76	0.71	0.90	0.95	0.85	0.65	0.71	0.00	0.85	0.69	0.53	0.45	1.12	0.97	0.74	1.10	1.09	0.89	0.77	1.04	1.07	1.12	1.01	0.02
Point #17	1.06	0.91	0.82	1.00	1.31	0.96	0.64	0.16	0.16	0.29	0.39	0.52	0.47	0.64	0.73	1.31	0.00	0.26	0.51	0.64	0.57	0.51	0.58	1.04	0.66	0.69	0.82	1.08	0.77	1.04	1.00	1.28
Point #18	1.03	0.91	0.72	0.86	1.28	1.03	0.69	0.41	0.18	0.13	0.44	0.55	0.44	0.46	0.57	1.06	0.26	0.00	0.26	0.39	0.73	0.57	0.46	1.03	0.75	0.64	0.69	1.03	0.81	1.04	0.95	1.03
Point #19	1.06	0.98	0.70	0.78	1.31	1.15	0.82	0.66	0.41	0.29	0.60	0.69	0.55	0.46	0.51	0.81	0.51	0.26	0.00	0.15	0.93	0.73	0.46	1.08	0.91	0.69	0.64	1.04	0.93	1.10	0.97	0.78
Point #20	1.10	1.04	0.72	0.77	1.34	1.22	0.91	0.78	0.53	0.41	0.71	0.78	0.64	0.41	0.53	0.69	0.64	0.39	0.12	0.00	1.04	0.82	0.51	1.13	1.00	0.75	0.66	1.07	1.00	1.15	1.01	0.66
Point #21	0.47	0.35	0.47	0.61	0.60	0.30	0.19	0.26	0.49	0.42	0.22	0.20	0.29	0.51	0.50	1.12	0.37	0.47	0.60	0.67	0.00	0.17	0.42	0.44	0.11	0.34	0.51	0.51	0.24	0.42	0.46	1.10
Point #22	0.37	0.26	0.31	0.45	0.53	0.30	0.08	0.26	0.43	0.30	0.09	0.03	0.13	0.34	0.33	0.97	0.33	0.37	0.47	0.53	0.17	0.00	0.26	0.35	0.11	0.19	0.34	0.40	0.17	0.34	0.35	0.95
Point #23	0.39	0.34	0.17	0.27	0.56	0.47	0.28	0.39	0.40	0.22	0.22	0.22	0.13	0.11	0.12	0.74	0.38	0.30	0.30	0.33	0.42	0.26	0.00	0.41	0.36	0.16	0.16	0.39	0.33	0.42	0.34	0.71
Point #24	0.07	0.14	0.42	0.53	0.27	0.32	0.43	0.95	1.16	0.90	0.66	0.53	0.59	0.72	0.61	1.69	1.04	1.03	1.08	1.13	0.67	0.54	0.63	0.00	0.49	0.39	0.51	0.15	0.31	0.06	0.15	1.66
Point #25	0.55	0.36	0.57	0.78	0.75	0.31	0.13	0.51	0.81	0.64	0.31	0.27	0.37	0.69	0.66	1.68	0.66	0.75	0.91	1.00	0.22	0.22	0.55	0.49	0.00	0.39	0.64	0.60	0.18	0.46	0.53	1.64
Point #26	0.35	0.29	0.17	0.41	0.64	0.49	0.26	0.64	0.78	0.51	0.33	0.24	0.22	0.36	0.29	1.37	0.69	0.64	0.69	0.75	0.53	0.29	0.24	0.39	0.39	0.00	0.26	0.39	0.29	0.41	0.28	1.33
Point #27	0.42	0.46	0.08	0.16	0.69	0.71	0.51	0.82	0.86	0.57	0.53	0.48	0.39	0.26	0.15	1.18	0.82	0.69	0.64	0.66	0.78	0.53	0.24	0.51	0.64	0.26	0.00	0.43	0.53	0.55	0.37	1.15
Point #28	0.07	0.24	0.35	0.40	0.27	0.47	0.51	1.01	1.17	0.90	0.70	0.59	0.61	0.67	0.54	1.60	1.08	1.03	1.04	1.07	0.78	0.61	0.59	0.15	0.60	0.39	0.43	0.00	0.42	0.21	0.08	1.56
Point #29	0.36	0.18	0.44	0.64	0.57	0.22	0.13	0.66	0.91	0.69	0.39	0.26	0.38	0.64	0.57	1.65	0.77	0.81	0.93	1.00	0.36	0.26	0.51	0.31	0.18	0.29	0.53	0.42	0.00	0.29	0.36	1.61
Point #30	0.12	0.06	0.45	0.57	0.29	0.27	0.41	0.94	1.16	0.91	0.65	0.52	0.59	0.75	0.64	1.73	1.04	1.04	1.10	1.15	0.64	0.53	0.64	0.05	0.46	0.41	0.55	0.21	0.29	0.00	0.20	1.69
Point #31	0.08	0.19	0.29	0.38	0.34	0.44	0.44	0.93	1.09	0.82	0.62	0.51	0.53	0.60	0.48	1.55	1.00	0.95	0.97	1.01	0.71	0.53	0.52	0.15	0.53	0.38	0.37	0.08	0.36	0.20	0.00	1.52
Point #32	1.61	1.60	1.24	1.18	1.82	1.82	1.53	1.43	1.14	1.06	1.36	1.42	1.27	0.97	1.05	0.04	1.28	1.03	0.78	0.66	1.69	1.46	1.10	1.66	1.64	1.33	1.15	1.56	1.61	1.69	1.52	0.00

Data32EG-3-280
Integrated Time Matrix

\# Pts: 32
Location: Egypt vic Grid 90..25..
Truck Type: 3 (777D Caterpillar Dump)
Elevation Factor: 280 (Total cuts and fills)

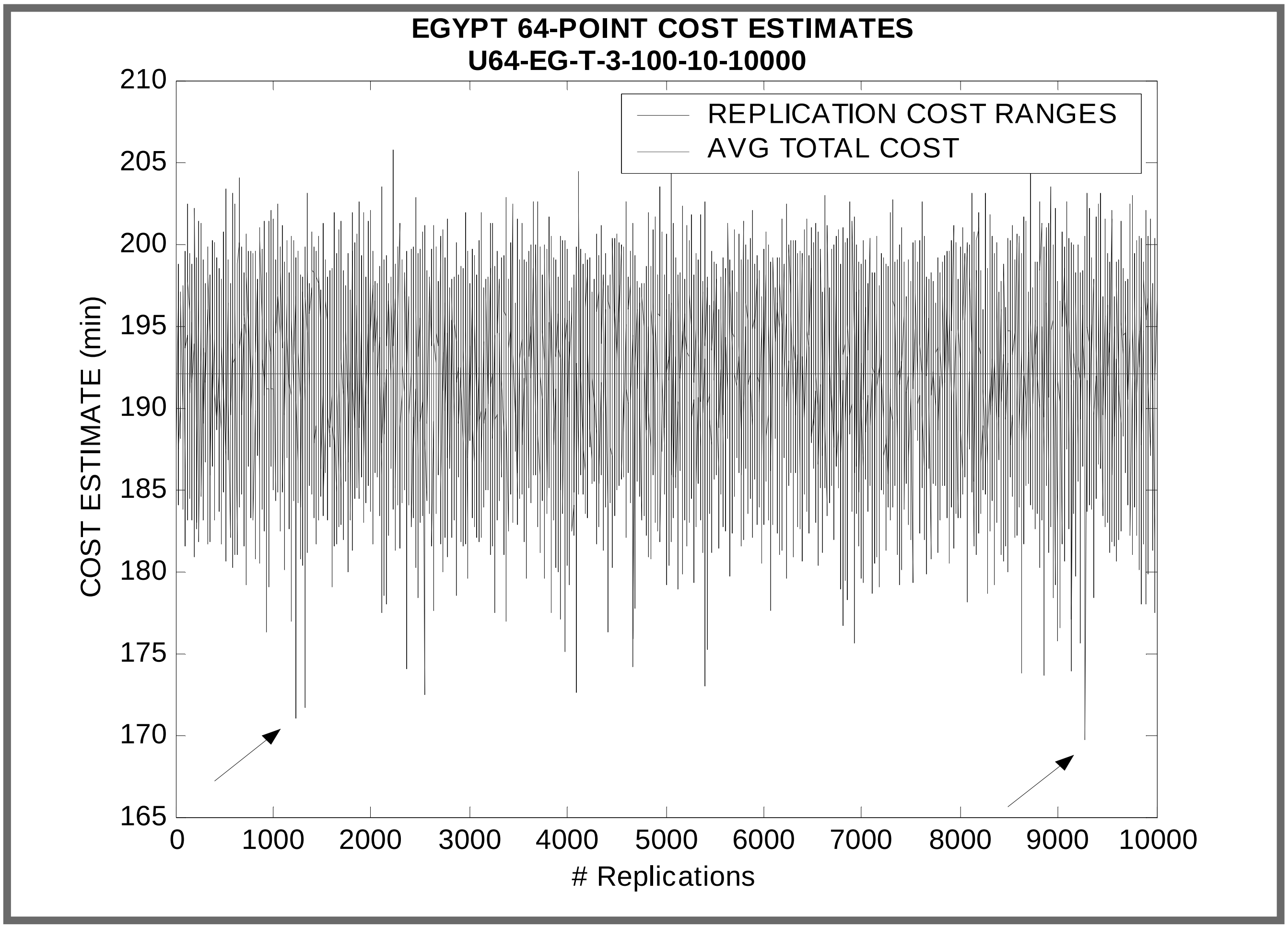
+20	+5	+5	+5	+5	+5	+5	+5	+5	+5	-10	-10	-10	-20	-20	+5	-10	+5	+5	-10	-10	-10	-10	+10	+5	-10	-10	+10	+5	+5	+10	+15

Data32EG-3-280a
Initial Vector of Relative Elevations

\# Pts: 32
Location: Egypt vic Grid 90..25..
Truck Type: 3 (777D Caterpillar Dump)
Elevation Factor: 280 (Total cuts and fills)

Howard D. McInvale
ISE Spring 2002

APPENDIX D


EGYPT 64-POINT COST ESTIMATES
U64-EG-T-3-100-10-10000
REPLICATION COST RANGES
AVG TOTAL COST
COST ESTIMATE (min)
Replications
210
205
200
195
190
185
180
175
170
165
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000

APPENDIX E

Computer Trials Log

Two Systems: PC (Dell P3) & Unix (Unix Sunblade)
Four Search Methods: Pure Local Search / Monte Carlo Search / Simulated Annealing / Threshold Accepting
Location: Egypt (vic grid 9025)
Vehicle Type: 3 (777D Caterpillar Dump)
Two Run Set-ups: 100 inner loops, 10 outer loops, 30 replications / 1000 inner loops, 10 outer loops, 3 replications

PC16-EG-L-3-100-10-30

Trial #	Time	Cost Range	Avg (std dev)
1	189	59-61	60 (0.4)
2	184	59-61	60 (0.6)
3	185	58-61	60 (0.7)
4	184	59-62	60 (0.7)
5	185	59-62	60 (0.8)
6	185	59-61	60 (0.7)
7	188	58-61	60 (0.6)
8	186	59-62	60 (0.9)
9	186	58-61	60 (0.7)
10	186	59-61	60 (0.7)
	185.8		60

PC16-EG-M-3-100-10-30

Trial #	Time	Cost Range	Avg (std dev)
1	188	65-67	66 (0.5)
2	186	64-67	66 (0.5)
3	187	65-66	66 (0.4)
4	186	65-67	66 (0.3)
5	187	65-66	66 (0.4)
6	186	64-67	66 (0.5)
7	185	65-67	66 (0.5)
8	185	64-67	66 (0.6)
9	185	64-67	66 (0.6)
10	184	65-67	66 (0.6)
	185.9		66

PC16-EG-S-3-100-10-30

Trial #	Time	Cost Range	Avg (std dev)
1	188	65-67	66 (0.6)
2	184	64-67	66 (0.5)
3	184	64-67	66 (0.5)
4	188	65-67	66 (0.7)
5	185	65-66	66 (0.4)
6	184	64-67	66 (0.6)
7	188	65-67	66 (0.4)
8	184	64-66	66 (0.5)
9	185	65-67	66 (0.6)
10	184	64-66	66 (0.4)
	185.4		66

PC16-EG-T-3-100-10-30

Trial #	Time	Cost Range	Avg (std dev)
1	185	65-67	66 (0.6)
2	187	64-67	66 (0.4)
3	186	65-67	66 (0.5)
4	185	64-66	66 (0.4)
5	185	64-67	66 (0.6)
6	184	65-67	66 (0.4)
7	183	64-66	66(0.4)
8	185	64-67	66 (0.6)
9	184	64-66	66 (0.4)
10	187	65-67	66 (0.5)
	185.1		66

U16-EG-L-3-100-10-30

Trial #	Time	Cost Range	Avg (std dev)
1	180	59-61	60 (0.5)
2	179	59-61	60 (0.6)
3	180	59-61	60 (0.7)
4	177	58-61	60 (0.7)
5	175	59-61	60 (0.7)
6	175	59-62	60 (0.7)
7	180	58-61	60 (0.6)
8	179	59-61	60 (0.4)
9	175	59-61	60 (0.6)
10	180	58-61	60 (0.6)
	178		60

U16-EG-M-3-100-10-30

Trial #	Time	Cost Range	Avg (std dev)
1	179	65-67	66 (0.4)
2	177	65-67	66 (0.6)
3	179	64-67	66 (0.5)
4	178	64-66	66 (0.4)
5	180	64-67	66 (0.5)
6	179	65-66	66 (0.5)
7	179	65-67	66 (0.4)
8	179	65-66	66 (0.3)
9	180	64-66	66 (0.5)
10	179	65-67	66 (0.5)
	178.9		66

U16-EG-S-3-100-10-30

Trial #	Time	Cost Range	Avg (std dev)
1	178	65-67	66 (0.4)
2	177	65-67	66 (0.5)
3	176	64-66	66 (0.4)
4	177	65-67	66 (0.4)
5	178	64-67	66 (0.4)
6	175	64-66	66 (0.4)
7	176	65-66	66 (0.4)
8	178	65-67	66 (0.4)
9	177	65-66	66 (0.2)
10	176	65-67	66 (0.5)
	176.8		66

U16-EG-T-3-100-10-30

Trial #	Time	Cost Range	Avg (std dev)
1	181	65-66	66 (0.3)
2	181	65-67	66 (0.4)
3	182	64-66	66 (0.4)
4	174	65-67	66 (0.5)
5	176	64-66	66 (0.3)
6	177	65-66	66 (0.4)
7	176	65-67	66 (0.5)
8	176	64-66	66 (0.2)
9	174	65-66	66 (0.4)
10	181	66-67	66 (0.4)
	177.8		66

PC16-EG-L-3-1000-10-3

Trial #	Time	Cost Range	Avg (std dev)
1	205	56-57	57 (0.2)
2	205	56-57	57 (0.1)
3	204	56-57	57 (0.2)
4	199	56-57	57 (0.1)
5	197	57-57	57 (0.2)
6	198	57-57	57 (0.1)
7	201	57-57	57 (0.3)
8	198	56-57	57 (0.2)
9	198	57-57	57 (0.4)
10	199	57-57	57 (0.1)
	200.4		57

PC16-EG-M-3-1000-10-3

Trial #	Time	Cost Range	Avg (std dev)
1	200	65-65	65 (0.1)
2	205	65-65	65 (0.1)
3	198	65-65	65 (0.2)
4	201	65-65	65 (0.2)
5	205	65-66	65 (0.4)
6	198	65-66	65 (0.4)
7	200	65-66	65 (0.4)
8	200	65-65	65 (0.3)
9	197	65-65	65 (0.2)
10	202	65-66	65 (0.5)
	200.6		65

PC16-EG-S-3-1000-10-3

Trial #	Time	Cost Range	Avg (std dev)
1	201	62-62	62 (0.1)
2	202	62-62	62 (0.2)
3	200	61-62	62 (0.2)
4	201	61-62	62 (0.4)
5	198	62-63	62 (0.4)
6	199	62-63	62 (0.2)
7	201	61-62	62 (0.3)
8	200	62-62	62 (0.2)
9	198	62-63	62 (0.2)
10	200	62-62	62 (0.1)
	200		62

PC16-EG-T-3-1000-10-3

Trial #	Time	Cost Range	Avg (std dev)
1	200	60-61	60 (0.5)
2	204	60-60	60 (0.6)
3	197	60-60	60 (0.2)
4	199	59-60	60 (0.2)
5	201	59-61	60 (0.2)
6	201	60-60	60 (0.3)
7	201	60-60	60 (0.4)
8	198	59-60	60 (0.3)
9	197	60-61	60 (0.2)
10	198	60-60	60 (0.2)
	199.6		60

U16-EG-L-3-1000-10-3

Trial #	Time	Cost Range	Avg (std dev)
1	200	57-58	57 (0.2)
2	201	57-57	57 (0.2)
3	201	56-57	57 (0.1)
4	202	57-57	57 (0.4)
5	202	57-57	57 (0.4)
6	200	57-58	57 (0.4)
7	198	56-57	57 (0.2)
8	201	57-57	57 (0.1)
9	203	56-57	57 (0.2)
10	202	57-57	57 (0.3)
	201		57

U16-EG-M-3-1000-10-3

Trial #	Time	Cost Range	Avg (std dev)
1	198	65-66	65 (0.5)
2	199	65-66	65 (0.1)
3	197	65-66	65 (0.4)
4	197	65-65	65 (0.2)
5	200	64-65	65 (0.3)
6	198	65-65	65 (0.4)
7	201	64-65	65 (0.3)
8	200	65-66	65 (0.6)
9	198	64-65	65 (0.3)
10	198	65-65	65 (0.2)
	198.6		65

U16-EG-S-3-1000-10-3

Trial #	Time	Cost Range	Avg (std dev)
1	198	61-62	62 (0.1)
2	200	62-63	62 (0.2)
3	197	62-63	62 (0.1)
4	197	62-63	62 (0.5)
5	197	62-62	62 (0.4)
6	198	62-62	62 (0.5)
7	200	62-63	62 (0.4)
8	198	62-62	62 (0.3)
9	198	61-62	62 (0.4)
10	197	62-62	62 (0.4)
	198		62

U16-EG-T-3-1000-10-3

Trial #	Time	Cost Range	Avg (std dev)
1	197	60-61	60 (0.2)
2	199	60-61	60 (0.3)
3	200	60-61	60 (0.2)
4	199	60-61	60 (0.2)
5	198	59-60	60 (0.5)
6	197	59-61	60 (0.2)
7	197	60-60	60 (0.3)
8	198	59-60	60 (0.2)
9	197	60-60	60 (0.3)
10	198	59-60	60 (0.1)
	198		60

Thesis Results Log Sheet

Field Data Runs: Head-to-Head (by method) PC vs UNIX

Goal: To evaluate model performance under greater loads...

Run Title	Avg Time (min)	Cost Range(hrs)	Avg (Std Dev)		Run Title	Avg Time (min)	Cost Range(hrs)	Avg (Std Dev)
PC32-EG-L-3-100-10-3	0.70	167-168	166 (4.8)		U32-EG-L-3-1000-10-3	0.62	167-168	168 (2.0)
PC32-EG-M-3-100-10-3	0.70	165-168	167 (1.5)		U32-EG-M-3-1000-10-3	0.62	168-169	168 (0.9)
PC32-EG-S-3-100-10-3	0.63	168-170	169 (1.3)		U32-EG-S-3-1000-10-3	0.62	167-170	169 (1.5)
PC32-EG-T-3-100-10-3	0.65	166-169	168 (1.8)		U32-EG-T-3-1000-10-3	0.62	168-171	170 (1.4)
PC32-EG-L-3-1000-10-3	6.02	165-166	166 (1.1)		U32-EG-L-3-1000-10-3	6.13	166-167	166 (3.1)
PC32-EG-M-3-1000-10-3	6.13	166-167	166 (0.5)		U32-EG-M-3-1000-10-3	5.97	164-167	165 (1.1)
PC32-EG-S-3-1000-10-3	6.17	163-166	165 (1.5)		U32-EG-S-3-1000-10-3	6.12	164-165	164 (0.8)
PC32-EG-T-3-1000-10-3	6.03	164-167	166 (1.5)		U32-EG-T-3-1000-10-3	6.17	166-168	167 (1.2)
PC32-EG-L-3-1000-10-10	20.20	166-167	166 (0.6)		U32-EG-L-3-1000-10-10	19.26	166-167	166 (0.3)
PC32-EG-M-3-1000-10-10	20.38	165-166	165 (0.8)		U32-EG-M-3-1000-10-10	19.01	164-165	165 (0.5)
PC32-EG-S-3-1000-10-10	20.41	163-164	164 (1.1)		U32-EG-S-3-1000-10-10	19.18	163-164	164 (0.6)
PC32-EG-T-3-1000-10-10	20.18	166-167	166 (1.2)		U32-EG-T-3-1000-10-10	19.36	166-167	166 (1.1)
PC32-EG-L-3-10,000-10-10	198.58	165-166	166 (0.7)		U32-EG-L-3-10,000-10-10	189.24	165-166	166 (1.0)
PC32-EG-M-3-10,000-10-10	200.60	163-164	163 (0.4)		U32-EG-M-3-10,000-10-10	189.68	163-163	163 (0.2)
PC32-EG-S-3-10,000-10-10	201.56	161-164	163 (1.4)		U32-EG-S-3-10,000-10-10	190.45	162-163	163 (0.7)
PC32-EG-T-3-10,000-10-10	198.63	164-165	164 (0.5)		U32-EG-T-3-10,000-10-10	189.62	164-165	164 (0.1)
PC32-EG-L-3-100,000-10-10	2000.62	165-166	166 (0.2)		U32-EG-L-3-100,000-10-10	1922.21	166-167	166 (1.2)
PC32-EG-M-3-100,000-10-10	2004.52	162-163	163 (0.8)		U32-EG-M-3-100,000-10-10	1918.16	163-164	163 (0.6)
PC32-EG-S-3-100,000-10-10	1999.52	160-161	161 (0.9)		U32-EG-S-3-100,000-10-10	1928.35	161-162	161 (0.4)
PC32-EG-T-3-100,000-10-10	1997.86	161-162	162 (1.1)		U32-EG-T-3-100,000-10-10	1930.02	161-162	162 (0.1)

Thesis Results Log Sheet

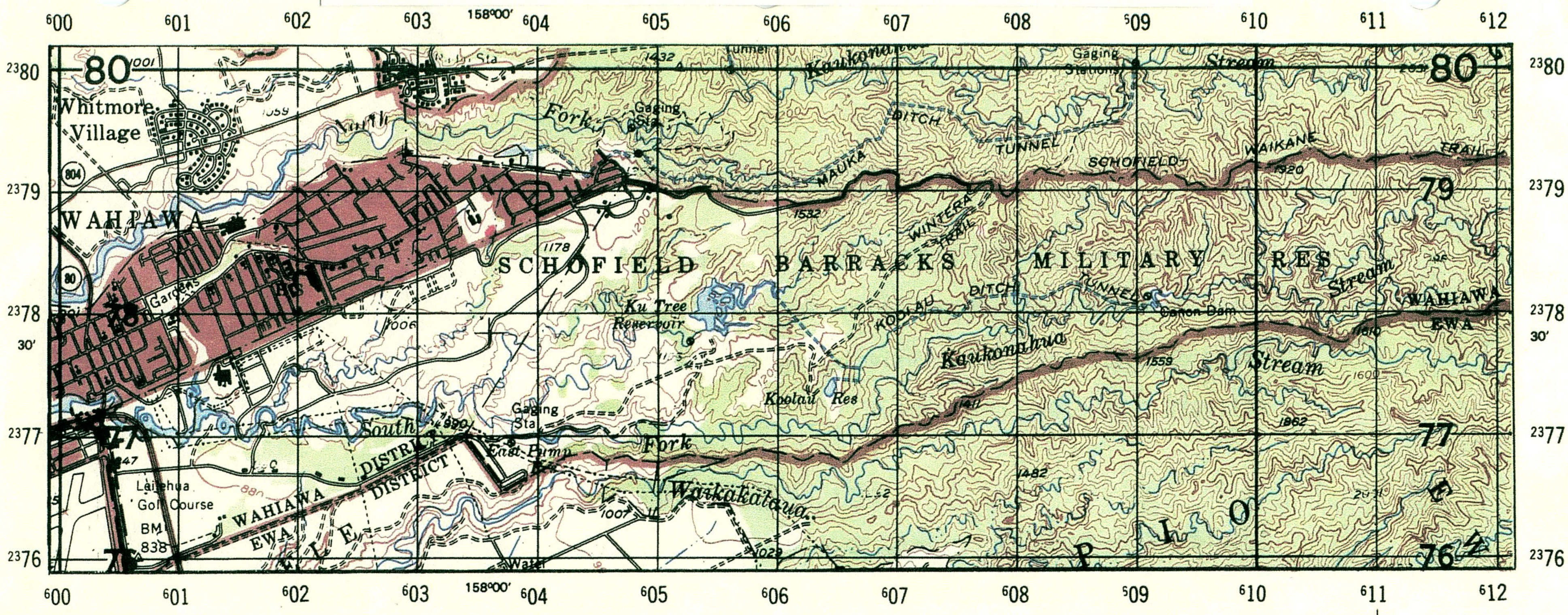
Field Data Runs: Head-to-Head (by method) PC vs UNIX

Goal: To evaluate model performance under greater loads...

Run Title	Avg Time (min)	Cost Range(hrs)	Avg (Std Dev)		Run Title	Avg Time (min)	Cost Range(hrs)	Avg (Std Dev)
PC32-EG-L-3-100-10-3	0.70	167-168	166 (4.8)		U32-EG-L-3-1000-10-3	0.62	167-168	168 (2.0)
PC32-EG-M-3-100-10-3	0.70	165-168	167 (1.5)		U32-EG-M-3-1000-10-3	0.62	168-169	168 (0.9)
PC32-EG-S-3-100-10-3	0.63	168-170	169 (1.3)		U32-EG-S-3-1000-10-3	0.62	167-170	169 (1.5)
PC32-EG-T-3-100-10-3	0.65	166-169	168 (1.8)		U32-EG-T-3-1000-10-3	0.62	168-171	170 (1.4)
PC32-EG-L-3-1000-10-3	6.02	165-166	166 (1.1)		U32-EG-L-3-1000-10-3	6.13	166-167	166 (3.1)
PC32-EG-M-3-1000-10-3	6.13	166-167	166 (0.5)		U32-EG-M-3-1000-10-3	5.97	164-167	165 (1.1)
PC32-EG-S-3-1000-10-3	6.17	163-166	165 (1.5)		U32-EG-S-3-1000-10-3	6.12	164-165	164 (0.8)
PC32-EG-T-3-1000-10-3	6.03	164-167	166 (1.5)		U32-EG-T-3-1000-10-3	6.17	166-168	167 (1.2)
PC32-EG-L-3-1000-10-10	20.20	166-167	166 (0.6)		U32-EG-L-3-1000-10-10	19.26	166-167	166 (0.3)
PC32-EG-M-3-1000-10-10	20.38	165-166	165 (0.8)		U32-EG-M-3-1000-10-10	19.01	164-165	165 (0.5)
PC32-EG-S-3-1000-10-10	20.41	163-164	164 (1.1)		U32-EG-S-3-1000-10-10	19.18	163-164	164 (0.6)
PC32-EG-T-3-1000-10-10	20.18	166-167	166 (1.2)		U32-EG-T-3-1000-10-10	19.36	166-167	166 (1.1)
PC32-EG-L-3-10,000-10-10	198.58	165-166	166 (0.7)		U32-EG-L-3-10,000-10-10	189.24	165-166	166 (1.0)
PC32-EG-M-3-10,000-10-10	200.60	163-164	163 (0.4)		U32-EG-M-3-10,000-10-10	189.68	163-163	163 (0.2)
PC32-EG-S-3-10,000-10-10	201.56	161-164	163 (1.4)		U32-EG-S-3-10,000-10-10	190.45	162-163	163 (0.7)
PC32-EG-T-3-10,000-10-10	198.63	164-165	164 (0.5)		U32-EG-T-3-10,000-10-10	189.62	164-165	164 (0.1)
PC32-EG-L-3-100,000-10-10	2000.62	165-166	166 (0.2)		U32-EG-L-3-100,000-10-10	1922.21	166-167	166 (1.2)
PC32-EG-M-3-100,000-10-10	2004.52	162-163	163 (0.8)		U32-EG-M-3-100,000-10-10	1918.16	163-164	163 (0.6)
PC32-EG-S-3-100,000-10-10	1999.52	160-161	161 (0.9)		U32-EG-S-3-100,000-10-10	1928.35	161-162	161 (0.4)
PC32-EG-T-3-100,000-10-10	1997.86	161-162	162 (1.1)		U32-EG-T-3-100,000-10-10	1930.02	161-162	162 (0.1)

Run Title	Run Time (min)	Best Cost (min)	Avg (Std Dev)
PC64-EG-S-3-100-10-10,000	3740.23 (2.5 days)	169 (at rep # 7,012)	192
U64-EG-T-3-100-10-10,000	3442.68 (2.5 days)	170 (at rep # 9035)	192
PC64-EG-S-3-100-10-1000	374.48 (6 hrs)	182	194
U64-EG-T-3-100-10-1000	343.59 (6 hrs)	182	194
PC64-EG-S-3-100-10-10	3.42	180	197
U64-EG-T-3-100-10-10	3.52	180	197
PC64-EG-S-3-1000-10-10	37.29	178	192
U64-EG-T-3-1000-10-10	34.86	178	192

1:5 00 # APPENDIX F 1985 OAHU SPECIAL EDITION EDITION 29EBT



Scale 1:50,000

1000	500	0	1000	2000	3000	4000	5000 Meters
1000	500	0	1000	2000	3000	4000	5000 Yards

PREPARED AND PRINTED SEPTEMBER 85 FROM THE OAHU SPECIAL ED 1-29EBT BY THE 29TH ENGR BN (T) FT SHAFTER HI 96858-5200 FOR THE 25TH INF DIV SCHOFIELD BKS, HI 96857-6000

CONTOUR INTERVAL 80 FEET
WITH SUPPLEMENTARY CONTOURS AT 20 FOOT INTERVALS

THIS MAP IS RED-LIGHT READABLE

LEGEND

Tint indicates built-up areas in which only landmark buildings are shown

LIMITED REVISION IN 1970

Hard surface, heavy duty road, four or more lanes wide — 4 LANES | 6 LANES
Hard surface, heavy duty road: Two lanes wide; Three lanes wide — 3 LANES
Hard surface, medium duty road, four or more lanes wide — 4 LANES | 6 LANES
Hard surface, medium duty road: Two lanes wide; Three lanes wide — 3 LANES
Improved light duty road, street
Unimproved dirt road Trail
Route markers Interstate ; State
Light, lighthouse, Windmill, wind pump
Buildings
Barns, sheds, greenhouses, etc.
Intermittent lake and stream
Mines: Open pit; Horizontal shaft; Vertical shaft; Prospect
Marsh or swamp; Dam
RAILROADS (Single track, Multiple track)
Standard gauge
Narrow gauge
In street
Car line
BOUNDARIES
National
State (with monument)
County
District
Corporate limits
Military reservation — MIL RES
Other reservation
Large rapids; Large falls
Rapids; Falls; Pier
Wrecks: Exposed; Sunken
Rocks: Sunken; Awash
Soundings in feet
Depth curves in feet
Foreshore flat
Limit of danger; Reef
Spot elevations in feet:
Checked — 168
Unchecked
Woods or brushwood
Bench mark, monumented — BM × 792

GRID ZONE DESIGNATION:
4Q

100,000 M. SQUARE IDENTIFICATION

EJ FJ
600

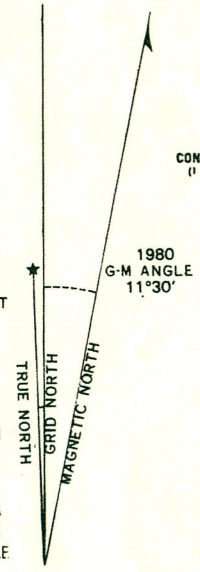
IGNORE the SMALLER figures of any grid number; these are for finding the full coordinates. Use ONLY the LARGER figures of the grid number; example: 2377000

TO GIVE A STANDARD REFERENCE ON THIS SHEET TO NEAREST 100 METERS

SAMPLE POINT: KOOLAU RES

1. Read letters identifying 100,000 meter square in which the point lies:	FJ		
2. Locate first VERTICAL grid line to LEFT of point and read LARGE figures labeling the line either in the top or bottom margin, or on the line itself:		06	
Estimate tenths from grid line to point:		4	
3. Locate first HORIZONTAL grid line BELOW point and read LARGE figures labeling the line either in the left or right margin, or on the line itself:			77
Estimate tenths from grid line to point:			
SAMPLE REFERENCE:	FJ064774		
If reporting beyond 18° in any direction, prefix Grid Zone Designation, as:	4QFJ064774		

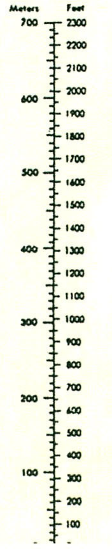
GRID CONVERGENCE 0°12′ (04 MILS) FOR CENTER OF SHEET



TO CONVERT A MAGNETIC AZIMUTH TO A GRID AZIMUTH ADD G-M ANGLE

TO CONVERT A GRID AZIMUTH TO A MAGNETIC AZIMUTH SUBTRACT G-M ANGLE

CONVERSION GRAPH
(1 meter = 3.28 feet)



References

Aarts E.H.L. and Lenstra J.K. *Local search in combinatorial optimization*. Wiley. New York. 1997.

Anily S. and R. Hassin. “The Swapping Problem.” *Networks* **22** (4): 419-433. JUL 1992.

Anily S. and G. Mosheiov. The Traveling Salesman Problem with Delivery and Backhauls. *Operations Research Letters* **16**. pp. 11-18. 1994.

Anily S. and J. Bramel. Approximation algorithms for the capacitated traveling salesman problem with pickups and deliveries. *Naval Research Logistics* **46**. pp. 654-670. 1999.

Bazaraa M.S. and J.J. Goode. "The traveling salesman problem: a duality approach." *Mathematical Programming* **13**. pp. 221-237. 1977.

Bellman R. "Dynamic programming treatment of the travelling salesman problem." *Journal of the Association for Computing Machinery* **9**, pp. 61-63. 1962.

Bock F. "An algorithm for solving traveling-salesman and related network optimization problems." *Research Report.* Armour Research Foundation. (Presented at the Operations Research Society of America Fourteenth National Meeting, St. Louis, October 24, 1958).

Carpaneto G. and M. Fischetti, and P. Toth. "New lower bounds for the symmetric traveling salesman problem." *Mathematical Programming* **45**. pp. 223-254. 1988.

Cirasella J. and D.S. Johnson, L.A. McGeoch and W. Zhang. The Asymmetric Traveling Salesman Problem: Algorithms, Instance Generators and Tests. *ALENEX 2001 Proceedings*. Springer Lecture Notes in Computer Science. 2001.

Croes G.A. "A method for solving traveling-salesman problems." *Operations Research* **6**. pp. 791-812. 1958.

Crowder H. and M.W. Padberg. "Solving large-scale symmetric travelling salesman problems to optimality." *Management Science* **26**. pp. 495-509. 1980.

Dantzig G. and R. Fulkerson, and S. Johnson. "Solution of a large-scale traveling-salesman problem." *Operations Research* **2**. pp. 393-410. 1954.

Flood M. M. "The traveling-salesman problem." *Operations Research* **4**. pp. 61-75. 1956.

Frieze and R. Karp, and B. Reed. “When is the assignment bound tight for asymmetric traveling salesman problem?” *SIAM Journal on Computing* v. **24** (June ’95) pp. 484-93.

Garey M.R. and D.S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. W. H. Freeman. San Francisco. 1979.

Glover F. and G. Gutin, A Yeo, and A. Zverovich. “Construction heuristics and domination analysis for the Asymmetric TSP.” *Proc. 3rd Workshop on Algorithm Engineering*. London, July 1999.

Gonzales R.H. "Solution to the traveling salesman problem by dynamic programming on the hypercube." *Technical Report* **18**. Operations Research Center, Massachusetts Institute of Technology. 1962.

Held M. and R.M. Karp. "A dynamic programming approach to sequencing problems." *Journal of the Society of Industrial and Applied Mathematics* **10**. pp. 196-210. 1962.

Heller I. "The travelling salesman's problem: part 1 -- basic facts." *Research Report*. George Washington University Logistics Research Project. 1954.

Heller I. "On the travelling salesman's problem." *Proceedings of the Second Symposium in Linear Programming (Volume 1)*. Washington, D.C. January 27-29. 1955.

Held M. and R.M. Karp. "The traveling-salesman problem and minimum spanning trees." *Operations Research* **18** (1970). pp. 1138-1162.

Held M. and R.M. Karp. "The traveling-salesman problem and minimum spanning trees: Part II." *Mathematical Programming* **1** (1971). pp. 6-25.

Henderson D. and R. Wakefield, D.E. Vaughan, and S.H. Jacobson. "Optimal Cut and Fill Strategies Using Local Search Algorithms." 2000.

Hong S. "A linear programming approach for the traveling salesman problem." Ph.D. Thesis, The Johns Hopkins University. 1972.

Jacobson S.H. and K.A. Sullivan and A.W. Johnson. Discrete manufacturing process design optimization using computer simulation and generalized hill climbing algorithms. *Engineering Optimization* **31** (1998). pp. 247-260.

Land, A. "The solution of some 100-city travelling salesman problems." Technical Report, London School of Economics. 1979.

Lin S. and B.W. Kernighan, "An effective heuristic algorithm for the traveling-salesman problem", *Operations Research* **21,** 498-516. 1973.

Karg R.L. and G.L. Thompson. "A heuristic approach to solving travelling salesman problems." *Management Science* **10**. pp. 225-248. 1964.

Kuhn H.W. "On certain convex polyhedra." Abstract 799t. *Bulletin of the American Mathematical Society* **61**. pp. 557-558. 1955.

Lin S. "Computer solutions of the traveling salesman problem." *The Bell System Technical Journal*. pp. 2245-2269. 1965.

Little J.D.C. and K.G. Murty, D.W. Sweeney, and C. Karel. "An algorithm for the traveling salesman problem." *Operations Research* **11**. pp. 972-989. 1963.

Martin G.T. "Solving the traveling salesman problem by integer linear programming." *Operations Research (Supplement 1)* **14**. Abstract WA7.10. 1966.

Menger K. "Das botenproblem." *Ergebnisse eines Mathematischen Kolloquiums* **2**. (K. Menger, editor). Teubner, Leipzig. pp. 11-12. 1932.

Miliotis P. "Integer programming approaches to the travelling salesman problem." *Mathematical Programming* **10**. pp. 367-378. 1976.

Miliotis P. "Using cutting planes to solve the symmetric travelling salesman problem." *Mathematical Programming* **15**. pp. 177-188. 1978.

Miller C.E. and A.W. Tucker and R.A. Zemlin. "Integer programming formulation of traveling salesman problems." *Journal of the Association for Computing Machinery* **7**. pp. 326-329. 1960.

Morton G. and A.H. Land. "A contribution to the travelling-salesman problem." *Journal of the Royal Statistical Society, Series B* **17**. pp. 185-194. 1955.

Ozan T. *Applied Mathematical Programming for Engineering and Production Management*. Prentice-Hall. Englewood Cliffs, NJ. 1986.

Padberg M.W. and S. Hong. "On the symmetric travelling salesman problem: a computational study." *Mathematical Programming Study* **12**. pp. 78-107. 1980.

Reiter S. and G. Sherman. "Discrete optimizing." *SIAM Journal on Applied Mathematics* **13**. pp. 864-889. 1965.

Rhee W. “Boundary Effects in the Traveling Salesman Problem.” *Operations Research Letters* **16** (1994) pp. 19-25. 1994.

Robacker J.T. "Some experiments on the traveling-salesman problem." *RAND Research Memorandum* RM-1521. 1955.

Robinson J.B. "On the Hamiltonian game (a traveling-salesman problem)." *RAND Research Memorandum* RM-303. 1949.

Schaufelberger John E. *Construction Equipment* Management. Prentice Hall. New Jersey 1999.

VITA

CPT Howard D. McInvale is a native of Birmingham, AL. Doug graduated from the United States Military Academy in 1993 and was commissioned into the infantry as a second lieutenant.

His engineering education includes a Bachelor of Science from West Point, in 1993 and a Master of Science in Industrial and Systems Engineering from Virginia Polytechnic Institute and State University (Virginia Tech), in 2002. His military education includes the Expert Infantryman Badge and completion of the Airborne, Air Assault, and Ranger schools.

Doug became a member of the faculty in the Department of Mathematical Sciences at the U.S. Military Academy at West Point in 2002.

Doug and his wife, Amy have two boys, Jackson and Joseph, and reside in West Point, NY.