Optimizing Earthwork Block Removal in Road Construction

by

Valentin Raphael Koch

B.Sc. Hons., The University of British Columbia, 2008

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The College of Graduate Studies

(Mathematics)

THE UNIVERSITY OF BRITISH COLUMBIA

(Okanagan)

April 2010

Abstract

In road construction, earthwork operations account for about 25% of the construction costs. Existing linear programming models for earthwork logistics optimization are designed to minimize the hauling costs and to balance the earth across the construction site. However, these models do not consider the removal of physical blocks that may influence the earthwork process.

In this thesis, we extend the linear programming model of Mayer and Stark [MS81] with the addition of a block removal schedule. The resulting model is a mixed-integer linear program. We analyze the model size and the schedule search space in order to make conclusion about the use of the model. Based on structural observations, we introduce a set of algorithms that significantly reduce the solving time of the model. Finally, we conduct numerical experiments to compare our solutions with the solutions of a traditional earthwork process that makes use of linear programming.

From our numerical results, we conclude that an optimal removal schedule produces solutions that are 4.1% cheaper on average than a traditional method, with savings that can go as high as 19%. We conclude our discussion with possible extensions to the model, that can help an engineer to design roads that are more economical and ecological with respect to the earthwork operations.

Table of Contents

Appendices

List of Tables

List of Figures

Glossary of notation

$a \in \mathcal{A}$	a is an element of set $\mathcal{A}$. 4
$\mathbb{R}$	Real numbers. 4
$\mathbb{R}_+$	Nonnegative real numbers. 5
$\phi : \mathcal{A} \to \mathcal{B}$	The mapping $\mathcal{A}$ into $\mathcal{B}$ by ϕ. 5
$[a, b]$	Interval $\{x \in \mathbb{R} : a \leq x \leq b\}$. 5
∞	Infinity. 5
f'	Derivative of function f. 6
$\int$	Integral. 5
$\lvert \cdot \rvert$	Absolute value. 7
$\lvert \mathcal{A} \rvert$	Cardinality of set $\mathcal{A}$. 7
$\{x : P(x)\}$	The set of all x such that $P(x)$ is satisfied. 8
$\mathbb{Z}_+$	Nonnegative integers. 8
$\mathcal{A} \cup \mathcal{B}$	The union of sets $\mathcal{A}$ and $\mathcal{B}$. 8
$\operatorname{dist}(i, j)$	Distance between section i and j. 8
$\mathbb{R}_{++}$	Positive real numbers. 8
$\min f$	The minimum of function f. 10
$\sum$	Summation. 10
$\mathcal{A} \times \mathcal{B}$	Cartesian product $\{(x, y) : x \in \mathcal{A}, y \in \mathcal{B}\}$. 12
$\wedge$	Logical connective representing *and*. 14
$\vee$	Logical connective representing *or*. 14
$\mathcal{A} \setminus \mathcal{B}$	Set difference $\{x : x \in \mathcal{A} \text{ and } x \notin \mathcal{B}\}$. 16
$\varnothing$	The empty set. 16
$\Theta\left(\cdot\right)$	Asymptotic tight bound. 22
$\mathcal{O}\left(\cdot\right)$	Asymptotic upper bound. 22
$\mathbb{Z}_{++}$	Positive integers. 22
$\mathcal{A} \subseteq \mathcal{B}$	$\mathcal{A}$ is a subset of $\mathcal{B}$. 22
$\binom{n}{k}$	Binomial coefficient (n choose k). 30
$\mathbb{R}^n$	Real n-vectors ($n \times 1$ matrices). 35
X^T	Transpose of matrix X. 35
$\mathbb{R}^{m \times n}$	Real $m \times n$ matrices. 38
$a \leftarrow b$	Assign value b to a. 37

Acknowledgements

Many people were involved in the creation of this thesis. First, I would like to express my sincere gratitude to my supervisors, Dr. Yves Lucet and Dr. Warren Hare, for their help and support throughout this research. I greatly enjoyed our lively discussions and conversations. It was a great privilege to have supervisors that helped me to overcome so many challenges, whether they were in mathematics, computer science, or simply personal.

I would also like to thank the members of my graduate committee, Dr. Heinz Bauschke and Dr. Shawn Wang, for their earnest support and guidance in my graduate studies, and for providing me insight into the beautiful subject of convex analysis.

It is a pleasure to thank my colleagues, especially Jim Nastos and Mason Macklem, for their patience to listen and the many ideas and input they contributed.

A very special thanks goes out to our industrial partner, for the professional assistance and valuable help on engineering questions.

I owe my deepest gratitude to my parents. Without their prayers and support, my academic career would have been impossible.

Finally, I would like to thank my wife Delphine, and my children Jordan and Tabitha, for their love and patience during these intense years.

This work was partially supported by the Ministry of Advanced Education of the Province of British Columbia through a Pacific Century Graduate Scholarship, and by the Mathematics of Information Technology and Complex Systems network through a MITACS Accelerate internship.

Dedication

To Delphine, Jordan, and Tabitha.

Chapter 1

Introduction

> La tendance la plus profonde de toute activité humaine est la marche vers l'équilibre.[1]
>
> Jean Piaget (1896–1980)

1.1 Road design

1.1.1 Road design and construction

Transportation of people and goods over land is, and has always been, an important part of civilization. Construction of the first roadways dates back as early as 5000 BCE [Lay99]. The invention of the wheel created the need for better roads and more sophisticated forms of road construction. Complex road design, including the alignment, earthwork, and pavement was already performed during the Roman Empire. The first scientific approach to road design was by Trésaguet in France at about 1764 [Lay99]; Trésaguet also introduced the first system of continuous road maintenance. The first comprehensive guide on road design and construction was published by McAdam in 1816 [McA24], introducing the concept of the elevation of roads through embankment. This resulted in much more stable roads, but also increased the cost of construction.

Roadways are an essential part of the economy in most countries. For example, in Canada in 2008, trucking accounted for 54% of the value of trade with the United States and commercial transportation services accounted for 4.1% of Canada's value-added GDP [Stu09]. At the same time, all levels of Canadian government combined spent $20.9 billion, or 1.6% of Canada's GDP, to build and maintain its 1 million kilometer road network. In view of these numbers, it is a natural consequence that governments of countries that depend heavily on road networks have a vast interest in keeping a

[1] The most profound tendency of all Human activity is progression toward equilibrium.

high quality road infrastructure, while at the same time trying to lower the expenditures on construction and maintenance. From an economic perspective, it is therefore important to try to minimize the cost and maximize the quality and safety of the road infrastructure.

The design of a road largely depends on safety requirements, which in turn depend on the traffic volume and vehicle speed. In 2007, in Canada, about 140,000 road collisions occurred, with about 16,000 seriously injured persons and about 3,000 fatalities. More than 100 million tonnes of dangerous goods were transported on Canadian roads during the same year [Stu09]. While the economical aspects in road construction are important, safety requirements are essential.

Depending on the location, the construction of a road can have a great impact on the environment. Maki [MKV02] shows that poorly planned road construction, especially in sensitive areas like the Amazon region, often harms the local economy instead of helping it. According to Maki, it is important to optimize the land use through environmental and social impact analysis. Appropriate classification and zoning of land should therefore be taken into account, and environmental impacts, like deforestation or earthwork, should be minimized when a road is built.

It follows from the above discussion that the road infrastructure needs to be optimized with respect to economic and social costs, safety requirements, and environmental impact.

1.1.2 Road design optimization

Multiple factors determine the total transportation cost of a road. Jha et. al. [JSJ06] list five main types of costs that affect the total transportation cost:

- planning, design, and administrative costs;
- construction costs;
- operation and maintenance costs;
- user costs; and
- social and environmental costs.

Planning and administrative costs are usually not considered in road design optimization. In [CGF89], Chew et. al. divide construction costs in six major categories, shown in Table 1.1 with an approximate associated

Table 1.1: Major components of construction costs

Pavement	30%
Earthwork (half of this is haul cost)	25%
Bridges	20%
Drainage	10%
Miscellaneous items	10%
Land	5%

percentage of the total construction cost. Operation and maintenance costs include roadway surface, roadside features, bridges, tunnels, snow and ice control, and traffic control devices and amount at about 5% of the construction costs, over a period of 30 years. The user costs, which include vehicle operating costs, travel time, and traffic accident costs, range from 300% to 1000% of the construction costs over a period of 30 years [JSJ06]. Finally, social and environmental costs can be the most critical factor in highway design. It is, however, difficult to quantify these costs and they are usually considered before the optimization of the design.

The design of the road can roughly be divided into geometric design and pavement design. The geometric design determines the alignment of the road and affects the construction costs, user costs, and social and environmental costs. The pavement design affects operation and maintenance, as well as user costs. Optimization of the geometric design is traditionally done in two stages: the horizontal alignment design and the vertical design.

The horizontal alignment is the road trajectory from a satellite's eye view. The primary objects that influence a horizontal alignment are forbidden areas like lakes, ocean shores, steep mountains, nature reserves, and private property. Cost parameters in horizontal alignments might include land costs and road distance. Early attempts at horizontal alignment optimization were based on dynamic programing [Tri87]. More recent approaches use genetic algorithms [JK06] or local neighborhood heuristics [LTL09].

Once the horizontal alignment is done, one can draw a line from the starting point to the end point of the road and look at the vertical ground profile along this line. The vertical alignment optimization tries to fit a road to the vertical ground profile as close as possible, while respecting grade constraints and other road specifications. The closer the road is to the ground profile, the less earthwork needs to be done in order to cut or fill sections of the road. Usually, vertical alignment optimization includes earthwork allocation and earth-moving optimization. In civil engineering, the vertical alignment optimization is commonly done through a manual iteration of computations that estimate the earthwork volumes for a given alignment. The alignment is then changed and the volumes are recomputed until a satisfying solution is found. Traditional mathematical models that automate the optimization of the vertical alignment use the method of *linear programming* (LP) [Eas88b, Mor96, Mor09, KL10].

Recently, the road alignment optimization research community has focused on the simultaneous optimization of horizontal and vertical alignment. In [CGF89], Chew et. al. modeled the road as a three-dimensional cubic spline. Jong and Schonfeld [JS03] use genetic algorithms to first solve for the three-dimensional alignment as a three-dimensional piecewise linear spline, and then fit circular curves to the horizontal and vertical projections of the spline. Parametric curves were studied by Makanae [Mak99] to model the road in three dimensions. Cheng and Lee [CL06] use a heuristic bi-level approach to iteratively optimize horizontal and vertical alignments.

All of the above optimization methods depend on the cost evaluation of the earthwork. In this thesis, we focus on the aspect of earthwork optimization for a given alignment. The optimal cost of earthwork is essential for the assessment of a given road alignment.

1.1.3 Earthwork optimization

Earthwork depends significantly on the terrain and the geometric design of the road. A road in a mountainous area requires more cutting and filling of earth than a road on the plain. An optimal vertical grade line minimizes the earthwork and satisfies all the design constraints for the road. To determine the optimality with respect to the earthwork, one needs to be able to compute the earthwork volumes for a given alignment.

In order to compute the earthwork volumes for a given grade line, cross-sections of the road profile are taken at regular intervals. The point along the horizontal axis where the cross-section is taken is called a *station*. Stations are usually arranged at an equal distance from one another along the horizontal axis of the grade line. The distance L_{ij} is the actual length between station i and j along the horizontal center line of the road. The area A_i of a cross-section at station i is determined and the volume $V \in \mathbb{R}$ between two stations i and j is computed as $V = L_{ij}(A_i + A_j)/2$. A traditional approach uses the *mass diagram* to depict the *net* accumulation of cut or fill between any two stations along the horizontal axis of the grade line. In [GH08], Garber and Hoel describe the mass diagram as follows.

> The ordinate of the mass diagram is the net accumulation

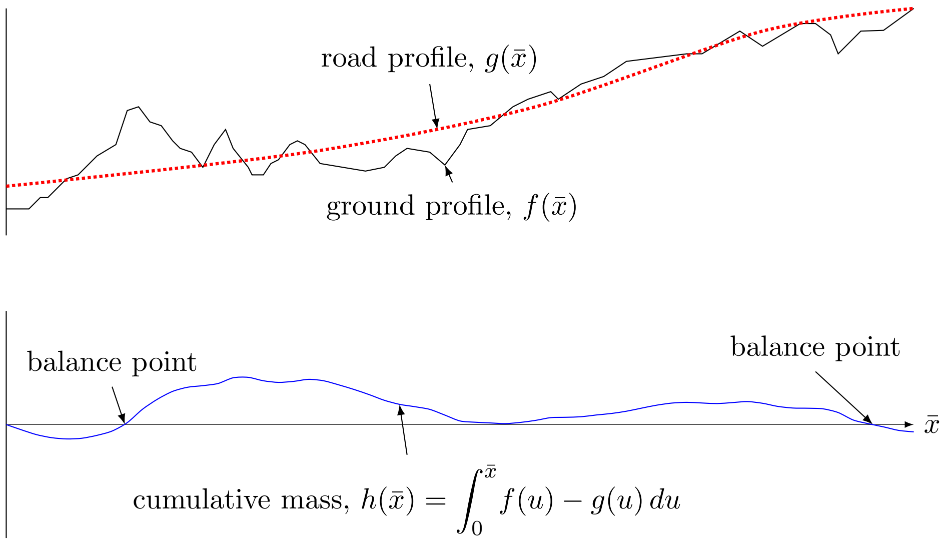


Figure 1.1: Example of vertical alignment with mass diagram

> in cubic meter from an arbitrary starting point. Thus, the difference in ordinates between any two stations represents the net accumulation of cut or fill between these stations. If the first station of the roadway is considered to be the starting point, then the net accumulation at this station is zero.

For this thesis, we will use the first station as the starting point. We can therefore define the mass diagram as follows.

Definition 1.1. Consider a vertical alignment. Let $\bar{x}$ be a station. If $f(\bar{x})$ represents the height of the ground profile at $\bar{x}$, and $g(\bar{x})$ represents the height of the road profile at $\bar{x}$, where $f : \mathbb{R}^+ \to \mathbb{R}$ and $g : \mathbb{R}^+ \to \mathbb{R}$ are piecewise continuous functions on $[0, +\infty[$, then the *mass diagram* is the graph of the Riemann integral

$$h(\bar{x}) = \int_0^{\bar{x}} (f(u) - g(u))\, du,$$

where $h(\bar{x})$ represents the cumulative earthwork mass at $\bar{x}$.

In Figure 1.1, we show an example of a vertical alignment with the corresponding mass diagram beneath. From Definition 1.1 and Figure 1.1, we observe the following characteristics.

- If $h'(\bar{x}) < 0$, then $\bar{x}$ is the position of a fill. If $h'(\bar{x}) > 0$, then $\bar{x}$ is the position of a cut.
- Any extremum of $h(\bar{x})$ represents a switch from cut to fill, or from fill to cut.
- The net accumulation between a station $\bar{x}_i$ and a station $\bar{x}_j$, where $\bar{x}_i < \bar{x}_j$ is $h(\bar{x}_j) - h(\bar{x}_i)$.
- For any two stations $\bar{x}_i$ and $\bar{x}_j$, if $h(\bar{x}_i) = h(\bar{x}_j)$, then the net accumulation between the two stations is zero. These points are called *balance points*, since they show points of balance between cut and fill volumes.
- A consequence of the last point is that if $h(0) = 0$ and $\bar{x}_{\text{end}}$ is the last station of the road, then an optimal alignment with respect to the balance of earth quantities has $h(\bar{x}_{\text{end}}) = 0$.

With the above observations, we can derive a *rule of thumb* to determine the hauling directions.

Remark 1.2. Let $h(0) = 0$. If $h(\bar{x}) > 0$ and $\bar{x}_i < \bar{x} < \bar{x}_j$, where $\bar{x}_i$ and $\bar{x}_j$ are balance points, then earth needs to be hauled ahead. The distance between $\bar{x}_i$ and $\bar{x}_j$ is the maximum push or haul distance. If $h(\bar{x}) < 0$ and $\bar{x}_i < \bar{x} < \bar{x}_j$, then earth needs to be hauled back. The values at the extrema represent the volumes of material that must be hauled along the road.

The mass diagram and the above rule of thumb have several shortcomings.

- The mass diagram does not take into account haul costs that are not proportional to the haul distance.
- The mass diagram does not take into account different soil characteristics.
- The mass diagram does not take into account *borrow pits* or *waste pits* that can be used to borrow additional quantities of soil or dispose of excessive material at off-the-roadway sites.
- The mass diagram does not take into account blocks like trees, rocks, and rivers that are present in the haul route.

To address some of the above shortcomings, Mayer and Stark [MS81] presented a LP model that not only balances the earth quantities, but also minimizes the hauling cost and incorporates soil material distribution, borrow pits, and waste pits. Based on [MS81], Easa developed a *mixed integer linear program* (MILP) that uses non-constant unit costs [Eas87], and a *quadratic program* (QP) that uses linear unit costs [Eas88a]. Jayawardane and Harris [JH90] extended the model in [MS81] by incorporating project duration and dynamic selection of borrow and waste pits. In this thesis, we extend a basic version of the model by Mayer and Stark, in order to overcome the problem of blocks in the haul route.

1.2 Basic linear programming model

In order to better explain our model of earth-moving, it is helpful to begin by considering the LP model in Mayer and Stark [MS81]. An extensive description of the model is given in [MS81] and [JH90]. We give a shorter outline in this section with some changes to the notation and refer the reader to the original paper for further details.

1.2.1 Notation

For the remainder of this thesis, the cardinality of a set $\mathcal{S}$ is denoted $|\mathcal{S}|$. The same notation $|\cdot|$ is used to indicate the absolute value and the meaning of the notation should be clear from the context.

1.2.2 Model parameters

In this section we provide information on the basic LP model parameters. Let a *section* be the region between two fixed stations along the horizontal axis. We divide the vertical alignment into m_s sections of equal length. Consider Definition 1.1. If for a section i, $\bar{x}_i$ represents the station at the beginning of the section, and $\bar{x}_{i+1}$ represents the station at the end of the section, then

$$h(\bar{x}) = \int_{\bar{x}_i}^{\bar{x}_{i+1}} (f(u) - g(u))\ du$$

is the *required change in volume* between the ground profile and the road profile in the section. The required change in volume for a section is positive if the section represents a cut, negative if the section represents a fill. As in [MS81, Eas87, JH90], we introduce m_b borrow pits and m_w waste pits in order to obtain a feasible solution in the case of an unbalanced vertical alignment. A borrow pit or a waste pit has a volume that represents the capacity. The following input parameters are therefore given:

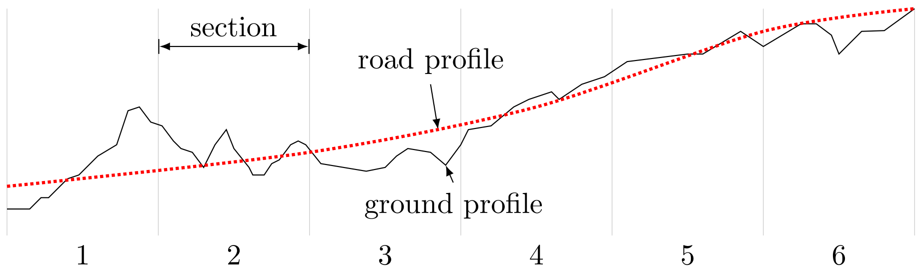


Figure 1.2: Example of vertical alignment with sections

- $m_s \in \mathbb{Z}_+$ is the number of sections in the road;
- $m_b \in \mathbb{Z}_+$ is the number of borrow pits;
- $m_w \in \mathbb{Z}_+$ is the number of waste pits.

To simplify notation, we introduce the following index sets:

- $\mathcal{S} = \{1, \ldots, m_s\}$ is the set of indices that is used to index sections;
- $\mathcal{B} = \{m_s + 1, \ldots, m_s + m_b\}$ is the set of indices that is used to index borrow pits;
- $\mathcal{W} = \{m_s + m_b + 1, \ldots, m_s + m_b + m_w\}$ is the set of indices that is used to index waste pits;
- $\mathcal{M} = \mathcal{S} \cup \mathcal{B} \cup \mathcal{W}$.

Finally, we need the required change in volume and the moving costs, which are defined by the following parameters:

- $V_i \in \mathbb{R}$ is the required change in volume of a section $i \in \mathcal{S}$ in cubic units, or the capacity of a borrow pit $i \in \mathcal{B}$ or waste pit $i \in \mathcal{W}$;
- $c_{ij} \in \mathbb{R}_{++} \cup \{+\infty\}$ is the cost of moving one cubic unit of material from i to j, where $i, j \in \mathcal{M}, i \neq j$.

In Figure 1.2, we show the alignment from Figure 1.1 that is divided into $m_s = 6$ sections of equal length. We denote the distance between section i and j as $\text{dist}(i, j)$, and evaluate this using

$$\text{dist}(i, j) = |\bar{x}_i - \bar{x}_j|, \tag{1.1}$$

where $\bar{x}_i$ is the midpoint of section i and $\bar{x}_j$ is the midpoint of section j.

Remark 1.3. The distance $\operatorname{dist}(i, j)$ can be defined arbitrarily. A more sophisticated definition than Equation (1.1) would incorporate the distance between the sections on the horizontal alignment as well as the difference of altitude. We leave this as a future research topic.

Note that there are many different ways to define the cost c_{ij} of moving earth. Examples of non-constant unit costs are given in [Eas87, Eas88a]. We use the following definition.

Definition 1.4. The cost $c_{ij} \in \mathbb{R}_{++} \cup \{+\infty\}$ to move one cubic unit of material from section i to section j, where $i, j \in \mathcal{M}$ and $i \neq j$, is defined as

$$c_{ij} = \begin{cases} c_{\text{exc}} + \operatorname{dist}(i,j)c_{\text{hau}} + c_{\text{emb}}, & \text{if } i, j \in \mathcal{S}, \\ c_{\text{bor}} + \operatorname{dist}(i,j)c_{\text{hau}} + c_{\text{emb}}, & \text{if } i \in \mathcal{B}, j \in \mathcal{S}, \\ c_{\text{exc}} + \operatorname{dist}(i,j)c_{\text{hau}} + c_{\text{was}}, & \text{if } i \in \mathcal{S}, j \in \mathcal{W}, \\ +\infty, & \text{otherwise,} \end{cases} \tag{1.2}$$

where $c_{\text{exc}} > 0$ is the unit cost of excavation, $c_{\text{hau}} > 0$ is the cost to haul one cubic unit of material over a unit distance, $c_{\text{emb}} > 0$ is the unit cost for embankment, $c_{\text{bor}} > 0$ is the unit cost to borrow material from a borrow pit, and $c_{\text{was}} > 0$ is the unit cost to dump material in a waste pit.

The following lemma follows clearly from Definition 1.4.

Lemma 1.5. *Consider any three sections $i, j, k \in \mathcal{S}$. If $i \neq k$, $i \neq j$, and $j \neq k$, then the strict triangle inequality $c_{ij} < c_{ik} + c_{kj}$ holds whenever $c_{ij} < +\infty$.*

The hauling distance from or to borrow pits or waste pits needs to account for dead-haul distances (the distance of the road from the pit-joining section to the pit). Excavation costs on a section are usually smaller than borrowing costs from a pit. This is usually balanced by the lower embankment costs of borrowed material. Also material that is dumped into a waste pit does not need a lot of embankment work and the embankment cost to dump material into a pit is much smaller than the embankment costs on a section.

1.2.3 Decision variables

The decision variable $x_{ij} \in \mathbb{R}$ represents the number of cubic units of earth that will be hauled from section i to section j, where $i, j \in \mathcal{M}$ and $i \neq j$.

1.2.4 Objective and constraints

The minimization of the earth-moving costs is modeled by LP (1.3) below

$$\min \sum_{\substack{i,j \in \mathcal{M} \\ i \neq j}} c_{ij} x_{ij} \tag{1.3a}$$

$$\text{s.t.} \sum_{\substack{j \in \mathcal{M} \\ j \neq i}} (x_{ij} - x_{ji}) = V_i \qquad \text{for all } i \in \mathcal{S}, \tag{1.3b}$$

$$\sum_{\substack{j \in \mathcal{M} \\ j \neq i}} x_{ij} \leq V_i, \qquad \text{for all } i \in \mathcal{B}, \tag{1.3c}$$

$$\sum_{\substack{j \in \mathcal{M} \\ j \neq i}} x_{ji} \leq V_i, \qquad \text{for all } i \in \mathcal{W}, \tag{1.3d}$$

$$x_{ij} \geq 0, \qquad \text{for all } i, j \in \mathcal{M}, i \neq j. \tag{1.3e}$$

Objective function

The objective function in (1.3a) minimizes the total hauling cost. Illogical moves (e.g., moving earth into a borrow pit) are penalized in Equation (1.2) of the cost Definition 1.4 by $+\infty$. In practice, to reduce memory size, decision variables that represent illogical moves are not implemented.

Constraints

For every road section i that has either a demand or a supply, we want the amount of incoming and outgoing earth movements to balance the demand or supply of the section. Constraints (1.3b) balance the demand and supply between sections and are called *balance constraints*. The model has $|\mathcal{S}|$ balance constraints.

We also need to account for the capacity of borrow and waste pits. We enforce the *capacity constraints* with inequalities (1.3c) and (1.3d). Since one constraint is needed for each borrow and waste pit, we have $|\mathcal{B}| + |\mathcal{W}|$ capacity constraints.

Finally, we can only move earth quantities that are greater or equal to zero. This is ensured by Equation (1.3e).

Observations

Remark 1.6. Note that shrink and swell factors that are mentioned in [MS81] are not included in this thesis since the model works with *banked units* and the mentioned factors are only taken into account to compute haul and compaction costs.

Proposition 1.7. *Let $V_s = \sum_{i \in \mathcal{S}} V_i$, $V_b = \sum_{i \in \mathcal{B}} V_i$, and $V_w = \sum_{i \in \mathcal{W}} V_i$. If $V_s \geq 0$ and $V_w - V_s > 0$, then the LP* (1.3) *has always a solution. If $V_s < 0$ and $V_b + V_s > 0$, then the LP* (1.3) *has always a solution.*

Proof. The conditions $V_w - V_s > 0$ or $V_b + V_s > 0$ ensure that a feasible point exists, as the waste or borrow pit provide sufficient slack. Since for all $i, j \in \mathcal{M}$, $c_{ij} \in \mathbb{R}_{++} \cup \{+\infty\}$ and $x_{ij} \geq 0$, we know that

$$\min \sum_{\substack{i,j \in \mathcal{M} \\ i \neq j}} c_{ij} x_{ij} \geq 0.$$

Hence, the LP (1.3) is bounded below. Since there exists a feasible point, and the LP is bounded below, there exists a solution to the LP (1.3). □

Proposition 1.8. *Let x^* be a feasible solution to the LP* (1.3)*. If $x^*_{ik} > 0$ and $x^*_{kj} > 0$, where $i \neq k$ and $k \neq j$, then x^* is not an optimal solution.*

Proof. If $x^*_{ik} \leq x^*_{kj}$, consider $\tilde{x}_{ik} = 0$, $\tilde{x}_{kj} = x^*_{kj} - x^*_{ik}$, and $\tilde{x}_{ij} = x^*_{ij} + x^*_{ik}$. Then

$$V_i = x^*_{ij} + x^*_{ik} + \sum_{\substack{s \in \mathcal{S} \\ s \neq i,j,k}} x^*_{is} - \sum_{\substack{s \in \mathcal{S} \\ s \neq i}} x^*_{si} = \tilde{x}_{ij} + \tilde{x}_{ik} + \sum_{\substack{s \in \mathcal{S} \\ s \neq i,j,k}} x^*_{is} - \sum_{\substack{s \in \mathcal{S} \\ s \neq i}} x^*_{si}, \tag{1.4}$$

$$V_k = x^*_{kj} - x^*_{ik} + \sum_{\substack{s \in \mathcal{S} \\ s \neq j,k}} x^*_{ks} - \sum_{\substack{s \in \mathcal{S} \\ s \neq i,k}} x^*_{sk} = \tilde{x}_{kj} - \tilde{x}_{ik} + \sum_{\substack{s \in \mathcal{S} \\ s \neq j,k}} x^*_{ks} - \sum_{\substack{s \in \mathcal{S} \\ s \neq i,k}} x^*_{sk}, \tag{1.5}$$

$$V_j = -x^*_{ij} - x^*_{kj} + \sum_{\substack{s \in \mathcal{S} \\ s \neq j}} x^*_{js} - \sum_{\substack{s \in \mathcal{S} \\ s \neq i,j,k}} x^*_{sj} = -\tilde{x}_{ij} - \tilde{x}_{kj} + \sum_{\substack{s \in \mathcal{S} \\ s \neq j}} x^*_{js} - \sum_{\substack{s \in \mathcal{S} \\ s \neq i,j,k}} x^*_{sj}, \tag{1.6}$$

and the volume constraints are satisfied. But from Lemma 1.5 we know that $c_{ik} + c_{kj} > c_{ij}$, and thus the total cost is cheaper, because

$$c_{ik}x^*_{ik} + c_{kj}x^*_{kj} + c_{ij}x^*_{ij} > (c_{ij} - c_{kj})x^*_{ik} + c_{kj}x^*_{kj} + c_{ij}x^*_{ij} = c_{ik}\tilde{x}_{ik} + c_{kj}\tilde{x}_{kj} + c_{ij}\tilde{x}_{ij}.$$

If $x^*_{ik} > x^*_{kj}$, consider $\tilde{x}_{ik} = x^*_{ik} - x^*_{kj}$, $\tilde{x}_{kj} = 0$, and $\tilde{x}_{ij} = x^*_{ij} + x^*_{kj}$. Then, the same arguments show that equations (1.4), (1.5), and (1.6) hold, and thus the volume constraints are satisfied, while the total cost is cheaper. □

Proposition 1.9. *The LP* (1.3) *contains* $|\mathcal{M}|^2 - |\mathcal{M}|$ *decision variables and* $|\mathcal{M}|^2$ *constraints.*

Proof. The number of moves from any section $i \in \mathcal{M}$ to any other section $j \in \mathcal{M}$ is equivalent to the number of pairs of section indecies, $(i, j), i, j \in \mathcal{M}$, that are formed by the Cartesian product $\mathcal{M} \times \mathcal{M}$. Now $|\mathcal{M} \times \mathcal{M}| = |\mathcal{M}|^2$. Since we do not count moves where $i = j$, we have $|\mathcal{M}|^2 - |\mathcal{M}|$ decision variables. The constraint in (1.3b) appears for every section in $\mathcal{S}$. Also, constraints (1.3c), and (1.3d) appear for each borrow pit in $\mathcal{B}$ and each waste pit in $\mathcal{W}$, respectively. Constraint (1.3e) appears for each of the $|\mathcal{M}|^2 - |\mathcal{M}|$ variables. It follows that we have

$$|\mathcal{S}| + |\mathcal{B}| + |\mathcal{W}| + |\mathcal{M}|^2 - |\mathcal{M}| = |\mathcal{M}|^2$$

constraints. □

Remark 1.10. In practice, since we do not implement illogical moves, the number of decision variables and constraints are less than the number given in Proposition 1.9. If we count only logical moves, we have $|\mathcal{S}|^2 - |\mathcal{S}|$ potential moves between sections, $|\mathcal{S}||\mathcal{B}|$ potential moves between sections and borrow pits, and $|\mathcal{S}||\mathcal{W}|$ potential moves between sections and waste pits. Thus in the actual implementation of the model, we have $|\mathcal{S}|(|\mathcal{M}| - 1)$ decision variables. Accordingly, we have

$$|\mathcal{S}| + |\mathcal{B}| + |\mathcal{W}| + |\mathcal{S}|(|\mathcal{M}| - 1) = |\mathcal{M}|(1 + |\mathcal{S}|) - |\mathcal{S}|$$

constraints in the actual implementation.

In the next chapter we will use these results in our discussion of the extension of the basic LP model (1.3).

Chapter 2

Mixed integer programming model

> He has barred my way with blocks of stone; he has made my paths crooked.
>
> Lamentations 3:9 (Holy Bible, NIV)

The model presented in Section 1.2 is a major improvement over the typical mass diagram due to the minimization of hauling distances and the incorporation of borrow and waste pits. However, in practice, a move from one section to another might not be possible due to natural blocks between the two sections. In this thesis, a block is defined as follows.

Definition 2.1. A *block* is an obstacle like a river, a group of trees, a rock-bed, or some other topographical feature that needs to be removed or otherwise dealt with before earth can be moved over the location of the block.

In a traditional earthwork construction process, clearing of blocks is usually treated as a separate task before excavation and embankment occurs [Trb90]. In such a case, the use of a mass diagram or a LP model, as the one described in Section 1.2, is perfectly valid, since after the obstacle removal, all the sections are accessible. This, however, does not account for earthwork that may occur during the obstacle removal (i.e. filling of a ditch or blasting of rock beds). During obstacle removal, stockpiles may be created and material may be borrowed elsewhere. This makes it difficult to assess the overall costs of obstacle removal, excavation, embankment, and haul in a traditional setting. Moreover, the order of removal of blocks might affect the final cost of the earthwork. We therefore extend the model in [MS81] with the introduction of blocks.

2.1 Notation

For the remainder of this thesis, the symbol $\wedge$ is a logical connective representing *and*, and the symbol $\vee$ is a logical connective representing *or*. Hence $A \wedge B$ means "A and B", where $A \vee B$ means "A or B"

2.2 Model parameters

We extend the basic LP formulation in Section 1.2 to develop a model that incorporates n_b blocks.

Definition 2.2. Let $\mathcal{I} = \{1, \ldots, n_b\}$. Each block is assigned to a road section. We let

$$\varsigma \colon \mathcal{I} \to \mathcal{S}$$

be the mapping that maps the block index to the assigned section index.

Also, we add the access roads that lead to the construction site of the road. Each access road is connected to one of the m_s sections. Access roads are typically attached to sections at the beginning and end of the road, but may also occur at any point along the road and in between blocks. The number and position of access roads influences the optimal schedule for the removal of the blocks.

Definition 2.3. Let $\mathcal{R} = \{1, \ldots, n_r\}$ be the set of all access roads. Each access road is assigned to a road section. We let

$$\varrho \colon \mathcal{R} \to \mathcal{S}$$

be the mapping that maps the access road index to the assigned section index.

Since each access road allows for transporting earth out to an external waste pit or importing earth from an external borrow site, we attach an external borrow and waste pit to every access road with infinite capacity each. These pits are accounted for in set $\mathcal{B}$ and $\mathcal{W}$.

We introduce discrete time steps to remove the blocks. The removal of a block requires one time step and the block is removed at the end of the time step. Multiple blocks may be removed within the same time step. In a worst-case scenario, where all the blocks need to be removed in sequential order, n_b time steps are needed to remove the blocks. Once all the blocks are removed, we may need an additional time step to finish all the earthwork. We therefore introduce $n_b + 1$ time steps t with $t \in \mathcal{T} = \{0, \ldots, n_b\}$.

The cost coefficients from Equation (1.2) remain the same in our model. However, we need a cost coefficient for a move for every time step. Hence, for each $t \in \mathcal{T}$ we set

$$c_{ijt} = c_{ij}.$$

Excavation and embankment costs on external pits should be higher than for sections.

Remark 2.4. If more than one access road is present, then any movement of earth between two sections could also go over the access roads. One possible way to allow such moves is to represent each access road as a section. Each section that has an access road attached could be *cloned* and the clone be added to the set $\mathcal{S}$. In this thesis, however, we do not account for such moves and we leave it as a further research topic.

In Section 1.2.2, we stated that the temporary stocking of earth between sections is avoided due to Lemma 1.5. With the introduction of blocks, however, this is not guaranteed anymore (as we demonstrate in Example 2.7). We therefore introduce for each section i a stockpile tolerance parameter $\epsilon_i^{\mathrm{stk}} \in \mathbb{R}_+$. The significance and usage of this parameter is shown in Section 2.4.5.

2.3 Decision variables

Similar to the cost coefficients, we keep the same variables representing the quantities of earth that are moved between sections. However, since we now have the flexibility of moving earth between any two sections at any time step, we multiply the variables in Section 1.2.3 by the number of time steps. The variable $x_{ijt} \in \mathbb{R}$ represents the quantity of earth that is moved from section i to section j at time step t.

A block k examined at time step t is represented with a binary variable. A zero value for a block shows that no or only partial work was done to remove the block during that time step, and a one indicates that the block was cleared at the end of time step t. Hence for block k with $k \in \mathcal{I}, t \in \mathcal{T}$, we define the variable that indicates the removal of block k at time t via

$$y_{kt} \in \{0, 1\},$$

such that if block k is still present at time t, then $y_{kt} = 0$.

2.4 Objective and constraints

The minimization of the earth-moving costs, including the removal of blocks, can therefore be stated as MILP (2.1) below

$$\min \sum_{t \in \mathcal{T}} \sum_{\substack{i,j \in \mathcal{M} \\ i \neq j}} c_{ijt} x_{ijt} \tag{2.1a}$$

$$\text{s.t.} \sum_{t \in \mathcal{T}} \sum_{\substack{j \in \mathcal{M} \\ j \neq i}} (x_{ijt} - x_{jit}) = V_i \qquad \text{for all } i \in \mathcal{S}, \tag{2.1b}$$

$$\sum_{t \in \mathcal{T}} \sum_{\substack{j \in \mathcal{M} \\ j \neq i}} x_{ijt} \leq V_i, \qquad \text{for all } i \in \mathcal{B}, \tag{2.1c}$$

$$\sum_{t \in \mathcal{T}} \sum_{\substack{j \in \mathcal{M} \\ j \neq i}} x_{jit} \leq V_i, \qquad \text{for all } i \in \mathcal{W}, \tag{2.1d}$$

$$\sum_{t=0}^{u} \sum_{\substack{j \in \mathcal{M} \\ j \neq \varsigma(k)}} (x_{\varsigma(k)jt} - x_{j\varsigma(k)t}) \geq y_{ku} V_{\varsigma(k)} \qquad \text{for all } k \in \mathcal{I}, u \in \mathcal{T}, V_{\varsigma(k)} \geq 0, \tag{2.1e}$$

$$\sum_{t=0}^{u} \sum_{\substack{j \in \mathcal{M} \\ j \neq \varsigma(k)}} (x_{\varsigma(k)jt} - x_{j\varsigma(k)t}) \leq y_{ku} V_{\varsigma(k)} \qquad \text{for all } k \in \mathcal{I}, u \in \mathcal{T}, V_{\varsigma(k)} < 0, \tag{2.1f}$$

$$x_{ijt} \leq M y_{k,t-1}, \qquad \begin{array}{l} \text{for all } t \in \mathcal{T} \setminus \{0\}, i, j \in \mathcal{M}, \\ k \in \mathcal{I}, \\ (i \leq \varsigma(k) \leq j) \vee (j \leq \varsigma(k) \leq i) \end{array} \tag{2.1g}$$

$$x_{ijt} \leq M y_{k,t-1} + M y_{k+1,t-1}, \qquad \begin{array}{l} \text{for all } t \in \mathcal{T} \setminus \{0\}, i, j \in \mathcal{M}, \\ k \in \mathcal{I} \setminus \{n_b\}, r \in \mathcal{R}, \\ \varsigma(k) \leq i, j \leq \varsigma(k+1) \wedge \{r : \\ \varsigma(k) \leq \varrho(r) \leq \varsigma(k+1)\} = \varnothing \end{array} \tag{2.1h}$$

$$\sum_{t=0}^{u} \sum_{\substack{j \in \mathcal{M} \\ j \neq i}} (x_{ijt} - x_{jit}) \leq V_i + \epsilon_i^{\text{stk}}, \qquad \text{for all } i \in \mathcal{S}, u \in \mathcal{T}, V_i \geq 0 \tag{2.1i}$$

$$\sum_{t=0}^{u}\sum_{\substack{j\in\mathcal{M}\\ j\neq i}}(x_{ijt}-x_{jit})\geq V_i-\epsilon_i^{\mathrm{stk}}, \qquad \text{for all } i\in\mathcal{S}, u\in\mathcal{T}, V_i<0 \tag{2.1j}$$

$$\sum_{t=0}^{u}\sum_{k\in\mathcal{I}} y_{ki}\geq u, \qquad \text{for all } u\in\mathcal{T}, \tag{2.1k}$$

$$x_{ijt}\geq 0, \qquad \text{for all } i,j\in\mathcal{M}, i\neq j, t\in\mathcal{T}, \tag{2.1l}$$

$$y_{kt}\in\{0,1\}, \qquad \text{for all } k\in\mathcal{I}, t\in\mathcal{T}, \tag{2.1m}$$

where $M\in\mathbb{R}$ is a large constant that exceeds the quantity of any possible earth move.

2.4.1 Objective function

As in the LP model, the MILP objective function (2.1a) minimizes the total hauling costs, but this time over all the time steps. Note that the cost coefficients for the block removal variables are set to zero.

2.4.2 Volume constraints

The balance constraints (2.1b) and the capacity constraints (2.1c) and (2.1d) have the same form as in the LP model, with an additional summation over all the time steps.

2.4.3 Block removal indicator

The decision variable y_{kt}, that indicates the removal of a block k at time step t, is binary. If we want to find an optimal removal schedule, we need to set some of these removal indicators to one. Once we decided to remove block k at time step t, we let $y_{kt}=1$. The removal of block k in a section $\varsigma(k)$ may involve earthwork. In order to effectively enforce the removal of the block at time step t, we need to satisfy any potential demand or supply of earth in this section at time t. For a supply, this is done by Equation (2.1e), for a demand, by Equation (2.1f). Therefore, the block removal indicator constraints relate the binary variables with the continuous variables.

Remark 2.5. Note that once a block k is removed at time step t, there is no need to enforce the monotonicity $y_{kt}\leq y_{k,t+1}$ in order to represent the status of a block in the later time steps. If we want to move earth over the block at some later time step $u > t$, we can just set the variables $y_{k,u-1}$ to 1 and the corresponding removal indicator constraint will already be satisfied. In fact, numerical tests showed that enforcing the monotonicity slows down the solving process.

2.4.4 Move feasibility constraint

Consider a move from section i to j. If there is a block k, such that $i \leq \varsigma(k) \leq j$, then we need to remove the block k before the move can happen. Therefore, if $y_{k,t-1} = 0$, then the move is not possible. However, if $y_{k,t-1} = 1$, then any (reasonable) quantity can be moved. In order to indicate any reasonable quantity, we use a very large constant M as upper bound for such a move. Equation (2.1g) models this relationship.

Consider a move from section i to j. If there are blocks k and $k+1$, such that $\varsigma(k) \leq i, j \leq \varsigma(k+1)$, and there is no access road between $\varsigma(k)$ and $\varsigma(k+1)$, then we need to remove either block k or block $k+1$ first, before the move can happen. This constraint is modeled in (2.1h).

The requirement that there is no access road in between $\varsigma(k)$ and $\varsigma(k+1)$ is shown in the condition of (2.1h) as $\{r : \varsigma(k) \leq \varrho(r) \leq \varsigma(k+1)\} = \varnothing$.

2.4.5 Stockpile constraints

Remark 2.6. Proposition 1.8 from the LP model does not apply to the block removal MILP model in (2.1). An explanation follows from Example 2.7 below.

Example 2.7. Consider the example with sections and volumes given in Table 2.1. The road section 3 represents a block with demand of one cubic unit of earth. Road section 4 has a supply of one, but is not accessible due to the block at section 3. The volume for road sections 1 and 2 is zero. There is one access road at the beginning of the road that joins section 1. For simplicity, let us consider $c_{\text{exc}} = c_{\text{hau}} = c_{\text{emb}} = c_{\text{bor}} = c_{\text{was}} = 1$. Since there is $n_b = 1$ block, we have 2 time steps. It is easy to see that the optimal solution to this problem is to first remove the block in section 3 by borrowing

Table 2.1: Road sections and volumes (bold font represents a block, ≈ an access road).

Section i	≈1	2	**3**	4
Volume V_i	0	0	-1	1

material from the adjacent section 2, followed by filling the newly created demand in section 2 with the cut of section 4. Hence the solution is

$$\begin{aligned} x_{230} &= 1 \\ x_{421} &= 1 \\ y_{10} &= 1, \end{aligned}$$

which is illustrated in Figure 2.1. The squares represent the road sections. The time steps are shown as the space between two layers of squares. A black colored square indicates the presence of a block, a gray colored square shows a block that was removed in a previous step. The arrows in between the layers represent earth movements. With the given cost model, any other

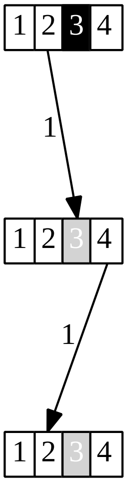


Figure 2.1: Earthwork scheme with removal schedule (black squares indicate blocks, gray squares indicate removed blocks).

solution would be more expensive.

If we disregard the time steps and add the quantities for each move over time, that is $x_{ij} = x_{ij0} + x_{ij1}$, for all $i, j \in \{1, 2, 3, 4\}$, then we have

$$\begin{aligned} x_{42} &= 1 \\ x_{23} &= 1 \end{aligned}$$

and the solution moves one cubic unit of earth from section 4 to 2, and from section 2 to 3. Hence, in this example, Proposition 1.8 does not apply to the MILP in (2.1).

From Remark 2.6, we conclude that a constraint is required to limit the amount of earth that can be stocked or borrowed from a road section. This can be enforced by a similar constraint like the balance constraint (2.1b), where the equality is replaced by a bound that represents the volume of a section plus or minus the stockpile tolerance. However, since stockpiling can occur at any time step, we need the constraint to hold at every time step. The resulting equations are (2.1i) for an upper bound, and (2.1j) for a lower bound.

2.4.6 Block removal enforcement

In order to provide the MILP solver with as much information as possible, we add the following constraint. At each time step $t = 1, 2, \dots, n_b$, we want to remove at least one block. This is guaranteed with constraint (2.1k). This constraint also ensures that no more than $n_b + 1$ time steps are needed.

2.4.7 Observations

Proposition 2.8. *Let $V_s = \sum_{i \in \mathcal{S}} V_i$, $V_b = \sum_{i \in \mathcal{B}} V_i$, and $V_w = \sum_{i \in \mathcal{W}} V_i$. If $V_s \geq 0$ and $V_w - V_s > 0$, then the MILP* (2.1) *has always a solution. If $V_s < 0$ and $V_b + V_s > 0$, then the MILP* (2.1) *has always a solution.*

Proof. The conditions $V_w - V_s > 0$ or $V_b + V_s > 0$ ensure that constraints (2.1b) to (2.1f), and (2.1i) to (2.1j) can be satisfied, since the waste or borrow pit provide sufficient slack. Since for all $i, j \in \mathcal{M}$, $t \in \mathcal{T}$, $c_{ijt} \in \mathbb{R}_{++} \cup \{+\infty\}$ and $x_{ijt} \geq 0$, we know that

$$\min \sum_{t \in \mathcal{T}} \sum_{\substack{i,j \in \mathcal{M} \\ i \neq j}} c_{ijt} x_{ijt} \geq 0.$$

In Section 4.1, we show how to set the binary variables y_{kt} in order to obtain a feasible point. Furthermore, in Section 3.2, we show that there are a finite number of ways to select the binary variables, and therefore a finite number of feasible points to the MILP (2.1).

Altogether, since the MILP (2.1) has a finite number of feasible points that are bounded below, and there exists at least one such point, there exists a solution to the MILP (2.1). □

Proposition 2.9. *The MILP* (2.1) *contains* $(|\mathcal{I}| + 1)(|\mathcal{M}|^2 - |\mathcal{M}|)$ *continuous, and* $(|\mathcal{I}| + 1)|\mathcal{I}|$ *binary decision variables.*

Proof. In order to get the number of continuous variables, we multiply the $|\mathcal{M}|^2 - |\mathcal{M}|$ variables from Proposition 1.9 with $|\mathcal{I}| + 1$ time steps. We have therefore $(|\mathcal{I}| + 1)(|\mathcal{M}|^2 - |\mathcal{M}|)$ continuous variables. Since there are $|\mathcal{I}|$ blocks and $|\mathcal{I}| + 1$ time steps, we have $(|\mathcal{I}| + 1)|\mathcal{I}|$ binary variables. Altogether, we have $(|\mathcal{I}| + 1)((|\mathcal{M}|^2 - |\mathcal{M}|) + |\mathcal{I}|)$ variables. □

Remark 2.10. As in Remark 1.10, the number of decision variables in the practical implementation of the model is smaller than the number given in Proposition 2.9. If we multiply the number of continuous variables from Remark 1.10 with $|\mathcal{I}|+1$ time steps, we get $|\mathcal{S}|(|\mathcal{M}|-1)(|\mathcal{I}|+1)$ continuous variables. The $(|\mathcal{I}|+1)|\mathcal{I}|$ binary variables remain the same as in Proposition 2.9.

In the next section we examine some of the properties of Model (2.1).

Chapter 3

Model analysis

> Aut inveniam viam aut faciam.[2]
>
> Hannibal (247–183 BCE)

3.1 Space complexity

In this section, we discuss the *space complexity* of Model (2.1). In particular, we look at the *growth rate* of the model size, with respect to the input parameters, that is required to store the model in the memory of a computer. In this context, we employ the usual asymptotic notation for space complexity. The following definition is a reminder of the most common asymptotic bounds that we are using.

Definition 3.1 (Cormen et al. [CLRS09, p. 43]). For a given function $g(n)$, we denote by $\Theta\left(g(n)\right)$ the set of functions

$$\begin{aligned}\Theta\left(g(n)\right) = \{f(n) :\ & \text{there exist } c_1 > 0, c_2 > 0, \text{ and } n_0 \in \mathbb{Z}_{++} \text{ such that} \\ & 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \text{ for all } n \geq n_0\},\end{aligned}$$

and we say that $g(n)$ is an *asymptotic tight bound* for $f(n)$. We denote by $\mathcal{O}\left(g(n)\right)$ the set of functions

$$\begin{aligned}\mathcal{O}\left(g(n)\right) = \{f(n) :\ & \text{there exist } c > 0, \text{ and } n_0 \in \mathbb{Z}_{++} \text{ such that} \\ & 0 \leq f(n) \leq c g(n) \text{ for all } n \geq n_0\},\end{aligned}$$

and we say that $g(n)$ is an *asymptotic upper bound* for $f(n)$ and $\Theta\left(g(n)\right) \subseteq \mathcal{O}\left(g(n)\right)$.

Lemma 3.2. *The growth rate of the total number of variables for the model in* (2.1) *is of order* $\Theta\left(|\mathcal{I}||\mathcal{M}|^2 + |\mathcal{I}|^2\right)$.

[2] I shall either find a way or make one.

Proof. From Proposition 2.9 we know that there are

$$(|\mathcal{I}|+1)(|\mathcal{M}|^2-|\mathcal{M}|)+(|\mathcal{I}|+1)|\mathcal{I}| = |\mathcal{I}||\mathcal{M}|^2+|\mathcal{I}|^2+|\mathcal{I}| \\ -|\mathcal{I}||\mathcal{M}|-|\mathcal{M}|^2-|\mathcal{M}|$$

variables. Let $|\mathcal{I}|=1, |\mathcal{M}|=2$. Clearly,

$$2(|\mathcal{I}||\mathcal{M}|^2+|\mathcal{I}|^2) \geq |\mathcal{I}||\mathcal{M}|^2+|\mathcal{I}|^2+|\mathcal{I}|-|\mathcal{I}||\mathcal{M}|-|\mathcal{M}|^2-|\mathcal{M}|,$$

and

$$\frac{1}{2}(|\mathcal{I}||\mathcal{M}|^2+|\mathcal{I}|^2) \leq |\mathcal{I}||\mathcal{M}|^2+|\mathcal{I}|^2+|\mathcal{I}|-|\mathcal{I}||\mathcal{M}|-|\mathcal{M}|^2-|\mathcal{M}|.$$

We therefore conclude that the growth rate for the size of the total variables is $\Theta\left(|\mathcal{I}||\mathcal{M}|^2+|\mathcal{I}|^2\right)$. □

Lemma 3.3. *The growth rate of the number of balance constraints* (2.1b)*, borrow pit capacity constraints* (2.1c)*, and waste pit capacity constraints* (2.1d) *is of order* $\Theta\left(|\mathcal{M}|\right)$*.*

Proof. Each section needs a balance constraint (2.1b). Hence there are $|\mathcal{S}|$ balance constraints. Each borrow pit and each waste pit needs a capacity constraint. Thus there are $|\mathcal{B}|+|\mathcal{W}|$ capacity constraints. Altogether, there are

$$|\mathcal{S}|+|\mathcal{B}|+|\mathcal{W}|=|\mathcal{M}|$$

balance and capacity constraints. The growth rate of the balance constraints is therefore of order $\Theta\left(|\mathcal{M}|\right)$. □

Lemma 3.4. *The growth rate of the number of removal indicator constraints* (2.1e) *and* (2.1f) *is* $\Theta\left(|\mathcal{I}|^2\right)$*.*

Proof. We have one block removal indicator constraint for each indicator variable. From Proposition 2.9 we know that the number of indicator variables is $|\mathcal{I}|(|\mathcal{I}|+1)$. Hence the number of indicator variable constraints is $|\mathcal{I}|(|\mathcal{I}|+1)$ which is in $\Theta\left(|\mathcal{I}|^2\right)$. □

Lemma 3.5. *The growth rate of the number of feasible move constraints* (2.1g) *and* (2.1h) *is of order* $\mathcal{O}\left(|\mathcal{M}|^2|\mathcal{I}|^2\right)$

Proof. We consider the $(|\mathcal{M}|^2 - |\mathcal{M}|)$ potential moves between all the sections over $|\mathcal{I}| + 1$ time steps. Each move needs at most a total of $|\mathcal{I}|$ feasible move constraints. If the move occurs over one or more blocks, we use constraint (2.1g) at most $|\mathcal{I}|$ times. If the move occurs in between two blocks, we use the constraint (2.1h) at most $|\mathcal{I}|$ times. If the move occurs over n blocks and between two blocks at the same time, we use constraint (2.1g) n times, and constraint (2.1h) at most $|\mathcal{I}| - n$ times. Altogether, we use a total of at most $|\mathcal{I}|$ constraints. We have therefore

$$(|\mathcal{M}|^2 - |\mathcal{M}|)(|\mathcal{I}| + 1)|\mathcal{I}|$$

feasible move constraints, which is of order $\mathcal{O}\left(|\mathcal{M}|^2|\mathcal{I}|^2\right)$. □

Lemma 3.6. *The growth rate of the number of stockpile constraints* (2.1i) *and* (2.1j) *is of order* $\mathcal{O}\left(|\mathcal{M}||\mathcal{I}|\right)$.

Proof. At each time step, we need two stockpile constraints for each section. Hence we have

$$2|\mathcal{T}||\mathcal{S}| = 2(|\mathcal{I}| + 1)|\mathcal{S}| \leq 2|\mathcal{I}||\mathcal{M}| + 2|\mathcal{M}|$$

stockpile constraints which is in $\mathcal{O}\left(|\mathcal{M}||\mathcal{I}|\right)$. □

Lemma 3.7. *The growth rate of the number of removal enforcement constraints* (2.1k) *is of order* $\Theta\left(|\mathcal{I}|\right)$.

Proof. At each time step, we need a constraint that sets the least number of blocks that must be removed at that time step. We therefore have $|\mathcal{T}| = |\mathcal{I}| + 1$ constraints which is in $\Theta\left(|\mathcal{I}|\right)$. □

Lemma 3.8. *The growth rate of the number of non-negativity constraints* (2.1l) *is of order* $\Theta\left(|\mathcal{M}|^2|\mathcal{I}|\right)$.

Proof. We need a non-negativity constraint (2.1l) for each of the continuous variables. From Proposition 2.9, we know that there are $(|\mathcal{I}|+1)(|\mathcal{M}|^2-|\mathcal{M}|)$ continuous variables, which is in $\Theta\left(|\mathcal{M}|^2|\mathcal{I}|\right)$. □

Lemma 3.9. *The growth rate of the number of binary constraints* (2.1m) *is of order* $\Theta\left(|\mathcal{I}|^2\right)$.

Proof. We need a binary constraint (2.1m) for each of the binary variables. From Proposition 2.9, we know that there are $(|\mathcal{I}| + 1)|\mathcal{I}|$ binary variables, which is in $\Theta\left(|\mathcal{I}|^2\right)$. □

Proposition 3.10. *The growth rate of the memory size for the MILP described in* (2.1) *is of order* $\mathcal{O}\left(|\mathcal{M}|^7\right)$. *If the number of blocks* $|\mathcal{I}|$ *is fixed, then the memory size for the MILP* (2.1) *is of order* $\mathcal{O}\left(|\mathcal{M}|^4\right)$.

Proof. We add the asymptotic bounds for all the constraints from Lemma 3.3 to Lemma 3.9. The growth rate of the sum of all the constraints is therefore

$$\Theta\left(|\mathcal{M}|\right)+\Theta\left(|\mathcal{I}|^2\right)+\mathcal{O}\left(|\mathcal{M}|^2|\mathcal{I}|^2\right)+\mathcal{O}\left(|\mathcal{M}||\mathcal{I}|\right)+\Theta\left(|\mathcal{M}|^2|\mathcal{I}|\right)+\Theta\left(|\mathcal{I}|^2\right), \tag{3.1}$$

which is of order $\mathcal{O}\left(|\mathcal{M}|^2|\mathcal{I}|^2+|\mathcal{I}|^2\right)$. From Lemma 3.2, we know that the number of variables is of order $\Theta\left(|\mathcal{I}||\mathcal{M}|^2+|\mathcal{I}|^2\right)$. Hence, the size of the problem is bounded by

$$\mathcal{O}\left(|\mathcal{M}|^2|\mathcal{I}|^2+|\mathcal{I}|^2\right)\times\Theta\left(|\mathcal{I}||\mathcal{M}|^2+|\mathcal{I}|^2\right)$$

which is of order $\mathcal{O}\left(|\mathcal{M}|^2|\mathcal{I}|^4+|\mathcal{M}|^4|\mathcal{I}|^3\right)$. Since $|\mathcal{B}|\geq 1$, and $|\mathcal{W}|\geq 1$, we know that $|\mathcal{I}|<|\mathcal{M}|$, and hence $|\mathcal{I}|\in\mathcal{O}\left(|\mathcal{M}|\right)$. The asymptotic upper bound for the memory size is therefore $\mathcal{O}\left(|M|^7\right)$. If we fix the number of blocks $|\mathcal{I}|$, the dimension of the problem is bounded by

$$\mathcal{O}\left(|\mathcal{M}|^2\right)\times\Theta\left(|\mathcal{M}|^2\right),$$

which is of order $\mathcal{O}\left(|\mathcal{M}|^4\right)$. □

Remark 3.11. The asymptotic bound of $\mathcal{O}\left(|M|^7\right)$ in Proposition 3.10 is a worst-case bound, where the number of blocks $|\mathcal{I}|$ could grow linearly with $|\mathcal{M}|$, the sum of all sections, borrow pits, and waste pits. We mentioned in Chapter 2 that $|\mathcal{I}|$ is typically between 5 and 20. Therefore if $|\mathcal{M}|$ is large, then $|\mathcal{I}|$ is usually small compared to $|\mathcal{M}|$. In that case, we can fix $|\mathcal{I}|$ and, as shown in Proposition 3.10, the memory size of the MILP (2.1) grows with order $\mathcal{O}\left(|\mathcal{M}|^4\right)$.

Regardless of the fact that we fix the number of blocks in practice, the memory growth rate in $\mathcal{O}\left(|\mathcal{M}|^4\right)$ remains a concern for roads with hundreds of sections. Most MILP solvers take advantage of the sparsity of (2.1) and the bound may not be attained. However, in our numerical tests, we still observed a growth rate that quickly exhausted the memory of a standard desktop computer. From the given bound and from practical experience using a 64 bit desktop computer with 8 GB of RAM and the Gurobi Optimizer 2.0 MILP solver, we recommend to keep the number of sections $|\mathcal{S}|\leq 200$.

3.2 Schedule search

In Section 3.1, we showed that the memory size grows in polynomial time with the input. However, the difficult part in our model is the scheduling for the block removal. In this section we analyze the scheduling problem more closely. In particular, we look at the schedule search space and the set of feasible schedules in the schedule search space.

For all time steps $t \in \mathcal{T}$, we list the variables y_{kt} as a binary string in the order of block position. Since $|\mathcal{T}| = |\mathcal{I}| + 1$, we have $|\mathcal{I}| + 1$ binary strings of length $|\mathcal{I}|$. At time $t = 0$, all the blocks are in place, thus $y_{k0} = 0$, and the binary string contains zeros only. At a time step $t > 0$, the corresponding binary string contains t or more ones that represent the removal of blocks. A binary string that contains ones only, represents a complete removal.

Definition 3.12. A *schedule* is a set of $|\mathcal{I}| + 1$ binary strings where each string belongs to a distinct time step.

A road with five blocks, for example, has an initial binary string of 00000. If one block is removed, say the first one to the left, then we have a new string which is 10000. We continue the removal until the binary string is 11111. Note that complete removal of all blocks may occur before time step $t = |\mathcal{I}| + 1$. In that case, the digits of the remaining $|\mathcal{I}| + 1 - t$ strings can be considered all 1 for all remaining time steps (the constraints affected by the binary variables for the remaining time steps are already satisfied).

Definition 3.13. The *schedule search space* is the number of potential schedules in the MILP (2.1) as defined by the binary selection. This includes infeasible schedules.

Lemma 3.14. *The schedule search space is asymptotically bounded by*

$$\mathcal{O}\left(2^{|\mathcal{I}|^2} 2^{|\mathcal{I}|}\right).$$

Proof. Since there are $|\mathcal{I}| + 1$ binary strings of length $|\mathcal{I}|$, we have

$$2^{|\mathcal{I}|(|\mathcal{I}|+1)}$$

potential schedules. It follows that the schedule search space is in

$$\mathcal{O}\left(2^{|\mathcal{I}|^2} 2^{|\mathcal{I}|}\right). \qquad \square$$

Table 3.1: Road sections and volumes (bold font represents a block, ≈ an access road).

Section i	**1**	2	**3**	**4**	≈5	**6**	**7**
Volume V_i	2	-2	-1	-1	5	-1	-2

00000
00100
00110
00111
01111
11111

(a) Feasible

00000
10000
10001
11001
11011
11111

(b) Infeasible

Figure 3.1: Potential schedules

Not all schedules in the schedule search space are feasible. Blocks can only be removed if they have an access road attached or if an adjacent block that is accessible is removed first. We are interested in the number of feasible schedules that lead to a complete removal.

Definition 3.15. The *feasible schedules* are the schedules that only remove a block if the block is accessible at the appropriate time. Hence, if $k \in \mathcal{I}$ is a block at time step $t \in \mathcal{T}$, then $y_{kt} = 1$ implies that either there exists an access road $r \in \mathcal{R}$, such that

$$\varsigma(k-1) \leq \varrho(r) \leq \varsigma(k) \vee \varsigma(k) \leq \varrho(r) \leq \varsigma(k+1),$$

or we have

$$y_{k-1,u} = 1 \vee y_{k+1,u} = 1, \qquad \text{for some } u \in \mathcal{T}, u < t.$$

Example 3.16. Consider the example with the sections given in Table 3.1. Sections 1, 3, 4, 6, and 7 are blocks. There is a single access road that joins section 5. Figure 3.1 shows two potential schedules. The schedule in Figure 3.1a is feasible, with a possible earthwork scheme shown in Figure 3.2. However, the schedule in Figure 3.1b is not feasible, since the blocks at either end of the road are not accessible. There are other blocks between the access road and the end of the road.

Blocks can be removed sequentially or simultaneously. From an access road with blocks on either side, there are three possible ways to remove

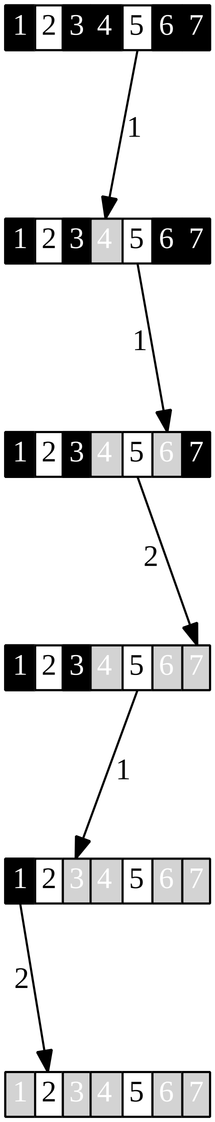


Figure 3.2: Earthwork scheme with removal schedule.

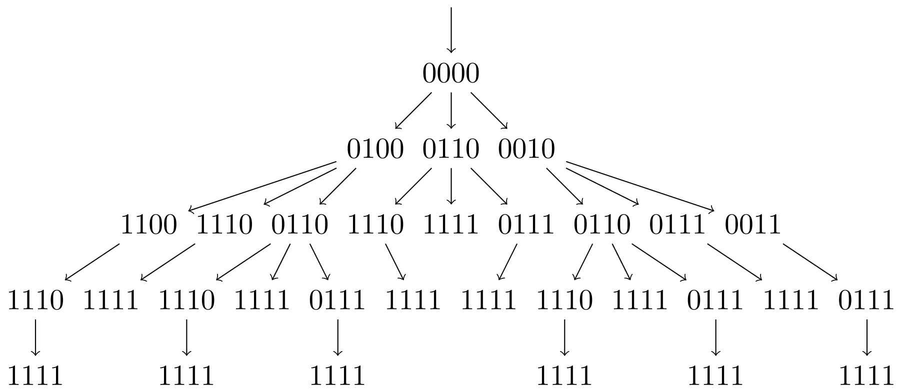
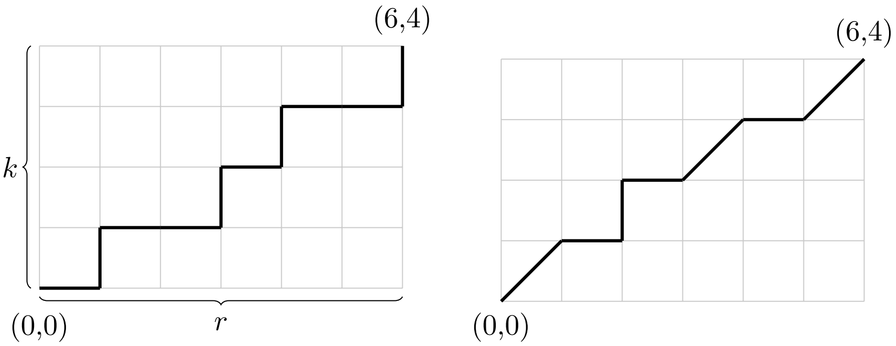


Figure 3.3: Tree structure of all feasible block removal schedules for 4 blocks with an access road joining between the two middle block

adjacent blocks. We can remove the block to the left, to the right, or both at once. If the access road is attached to the block, the block needs to be removed first. Using edges to represent left, right, or simultaneous removal, we can draw a tree that contains all the feasible schedules for a given configuration of access points. An example for 4 blocks with an access road attached to a section between the two center blocks is shown in Figure 3.3.

We start by approximating the number of all feasible schedules for a single access road from below. We therefore restrict the set of feasible schedules by analyzing schedules that allow left and right removal only, using one single access road. Let the access road be at a section between block r and block $k = |\mathcal{I}| - r$. Consider the $(k \times r)$-lattice of integral points in $\mathbb{Z}^2$ with coordinate (0,0) at the lower left corner and coordinate (k,r) at the upper right corner. We associate a left block removal to a step upward and a right block removal to a step to the right. We call the vertical steps (1,0)-steps and the horizontal steps (0,1)-steps to represent the increase of the $y-$ and $x-$axis respectively. A lattice path consisting of (0,1)- and (1,0)-steps that starts at (0,0) and ends at (k,r) is in bijection with a schedule starting at the root node with a zero binary string and ending at a leaf with a binary string of ones. Figure 3.4a depicts such a path. The next Lemma provides a formula to count all the possible lattice paths starting at (0,0) and ending at (k,r). We provide the proof as outlined in [Aig07].

Lemma 3.17 (Aigner [Aig07, p. 17]). *Consider a $(k \times r)$-lattice of integral points in $\mathbb{Z}^2$. The number of paths consisting of (1,0) and (0,1) steps from*



(a) (1,0),(0,1)-lattice path

(b) Delannoy path

Figure 3.4: (6×4)-lattice in $\mathbb{Z}^2$, representing 10 block configuration with a single access road between block 6 and 4

(0,0) to (k,r) is

$$L(k,r) = \binom{k+r}{r}. \tag{3.2}$$

Proof. Assign the symbol L(eft) to a (1,0)-step and R(ight) to a (0,1)-step. Every lattice path is in bijection with a $(k+r)$-word over $\{L, R\}$ containing exactly r L's. This corresponds to $\binom{k+r}{r}$ possible combinations of $(k+r)$ words containing r number of L symbols. □

Remark 3.18. For the block removal schedules, since $k + r = |\mathcal{I}|$, for $|\mathcal{I}|$ blocks and an access road between block k and r, we have

$$L(k,r) = \binom{|\mathcal{I}|}{r} \tag{3.3}$$

feasible schedules for the restricted set of feasible schedules for a single access road that contain left and right removals only.

Remark 3.19. Equation (3.3) is the formula for the *binomial coefficients* [Aig07, p. 16]. Consider the first five rows of *Pascal's triangle*

$$\begin{array}{ccccccccc}
 & & & & 1 & & & & \\
 & & & 1 & & 1 & & & \\
 & & 1 & & 2 & & 1 & & \\
 & 1 & & 3 & & 3 & & 1 & \\
1 & & 4 & & 6 & & 4 & & 1.
\end{array}$$

Let the rows represent the number of blocks $|\mathcal{I}|$, and the columns represent block r, where a single access road is between block k and r. The binomial coefficients in the triangle represent the number of feasible left and right removal schedules. We notice that the *central binomial coeffcients* along the central axis of the triangle grow the fastest. They correspond to an access road between k and r, where $k = r$. If $k = r$, with the formula given in [Wei02, p. 369], then

$$L(r,r) = \binom{2r}{r} = \binom{|\mathcal{I}|}{r}.$$

Using Stirling's approximation [Wei02, p. 2868] for the factorial function, we have

$$\binom{|\mathcal{I}|}{r} \leq \frac{4^r}{\sqrt{3r+1}}, \qquad \text{for all } r \geq 1.$$

Since $|\mathcal{I}| = 2r$, we conclude that the number of feasible schedules for a single access road and left and right removal only is in $\mathcal{O}\left(2^{|\mathcal{I}|}\right)$.

We now extend our restricted set of feasible schedules by adding another removal step to the schedules with one access. So far we considered left-removal and right-removal steps only. But if the access road is in between two blocks, we can also remove the left and right block at the same time, as shown in Figure 3.3. Remark 3.19 is therefore a lower bound on the worst-case. Instead of having only (0,1)- and (1,0)-steps in our lattice path, we now add a diagonal (1,1)-step. This results in *Delannoy* paths as shown in Figure 3.4b. The proof for the next Lemma is already given in [SAV$^+$03]. We prove it in a slightly different way.

Lemma 3.20. *Consider a $(k \times r)$-lattice of integral points in $\mathbb{Z}^2$. The number of Delannoy paths consisting of (1,0),(0,1) and (1,1) steps from (0,0) to (k,r) is*

$$D(k,r) = \sum_{i=0}^{k} \binom{k}{i} \binom{k+r-i}{k}. \tag{3.4}$$

Proof. Assign the symbol L(eft) to a (1,0)-step, R(ight) to a (0,1)-step, and B(oth) to a (1,1)-step. Every Delannoy path corresponds therefore to a string in the $\{L, R, B\}$ alphabet. Consider a Delannoy path with i diagonals. Hence the corresponding string contains $r-i$ L's and $k-i$ R's. Now project[3] the (1,1)-steps onto the horizontal x-axis. The projected $\{L, R\}$ string has

[3] The projection of the diagonals onto a lattice is based on an idea by Jim Nastos.

now $r-i$ L's and $k-i+i=k$ R's. Using the formula for (1,0),(0,1)-lattice paths in Lemma 3.17, Equation (3.2), we have

$$L(k,r-i)=\binom{k+r-i}{r-i}=\binom{k+r-i}{k} \tag{3.5}$$

(1,0),(0,1)-paths. However, there are $\binom{k}{i}$ possible ways we can choose the diagonals that we project. Also, we have $i\in\{0,\ldots,k\}$. Hence we multiply Equation (3.5) by $\binom{k}{i}$ and sum i from 0 to k in order to get the formula in Lemma 3.20. □

Proposition 3.21. *The number of feasible schedules for a road with a single access road on either end of the road is in* $\mathcal{O}(1)$*.*

Proof. We use Equation (3.4) with $k=|\mathcal{I}|, r=0$ and obtain $D(|\mathcal{I}|,0)=1$. □

We are now able to compute the exact number of feasible schedules for a given road with one access road. However, a lot of construction sites have more than one access road. We are particularly interested in the very common case of two access roads, one at the start and one at the end of the road.

Lemma 3.22. *The number of feasible schedules for a road with two access points, one at the start and one at the end of the road, is*

$$S(|\mathcal{I}|)=\sum_{j=1}^{|\mathcal{I}|}D(|\mathcal{I}|-j,j), \tag{3.6}$$

where D is defined in Equation (3.4)*.*

Proof. Consider a binary string with $|\mathcal{I}|$ zeros that represents the time step at time $t=0$. If we remove a block at the start of the road, we set the left-most digit from 0 to 1. If we remove a block at the end, we set the right-most digit to 1. Likewise, if we simultaneously remove on both sides, we set the outer-most digits to 1. Since there is no access point anywhere else, this procedure continues until all digits are set to 1. The last digit or the last two digits that are set to 1 represent the *meeting point* of the construction crews, once all the blocks are removed. For any removal schedule with a meeting point $j\in\{1,\ldots,|\mathcal{I}|\}$, we can invert the schedule in the sense that zeros are set to one, and ones to zero. A inverted schedule represents a schedule that has the meeting point as a single access road. For a fixed meeting point, there is a bijection between all the schedules with two access roads that meet at the meeting point and the schedules with a single access road attached to the meeting point. In order to get all feasible schedules, we consider all $|\mathcal{I}|$ possible meeting points. Hence we sum j over $\{1, \dots, |\mathcal{I}|\}$ and compute the inverse schedules with (3.4), where $r = j$ and $k = |\mathcal{I}| - j$. This leads to the formula in Lemma 3.22. □

As presented in Banderier and Schwer [BS05], the *central Delannoy numbers* along the diagonal from (0,0) to any (r,r) grow the fastest. To get a worst-case scenario for a configuration with $|\mathcal{I}|$ blocks and one access road, we let the access road join the center of the binary string, thus $k = r = |\mathcal{I}|/2$. In the same paper [BS05], Banderier and Schwer also give an approximation to compute an upper bound on the central Delannoy numbers. It is given as

$$\begin{aligned} D(k,k) = 5.82842709^k (0.57268163k^{-1/2} &- 0.06724283k^{-3/2} \\ &+ 0.00625063k^{-5/2} + \mathcal{O}\left(k^{-7/2}\right)). \quad (3.7) \end{aligned}$$

Proposition 3.23. *The number of feasible schedules for a road with a single access road between the center blocks of the road is in $\mathcal{O}\left(3^{|\mathcal{I}|}\right)$.*

Proof. Consider Equation (3.4). If the single access road is between the center blocks, we have $r = k$. The dominating term in the sum of Lemma 3.22 is the number for the central Delannoy paths $D(k,k)$ when $k = |\mathcal{I}|/2$. We therefore use Equation (3.7) as an approximation. Since

$$5.83^{|\mathcal{I}|/2}(|\mathcal{I}|/2)^{-1/2} \lesssim 2.42^{|\mathcal{I}|}\sqrt{2}|\mathcal{I}|^{-1/2},$$

we conclude that the number of feasible schedules is in $\mathcal{O}\left(3^{|\mathcal{I}|}\right)$. □

Proposition 3.24. *The number of feasible schedules for a road with two access roads, one on each end of the road, is in $\mathcal{O}\left(3^{|\mathcal{I}|}\right)$.*

Proof. Use Equation (3.6) with $|\mathcal{I}| = 0, 1, 2, \dots$. We obtain the sequence $\{0, 1, 2, 5, 12, 29, 70, 169, \dots\}$. These are the *Pell-Lucas* numbers [Wei02, p. 2185], given by the closed form

$$P(n) = \frac{(1+\sqrt{2})^n - (1-\sqrt{2})^n}{2\sqrt{2}}.$$

Since $P(n) \leq (1+\sqrt{2})^n\sqrt{2}/4 \approx 2.42^n\sqrt{2}/4$, we conclude that the number of feasible schedules is in $\mathcal{O}\left(3^{|\mathcal{I}|}\right)$. □

Remark 3.25. Consider Proposition 3.21, 3.23, and 3.24. If there is one access road only and the number of blocks $|\mathcal{I}|$ is small, then the number of feasible schedules is clearly tractable. In this case, it is reasonable to enumerate all the feasible schedules and to solve a LP for each of the feasible schedules. The optimal removal schedule is the schedule for the LP solution with the lowest objective value. The enumeration of the feasible schedules can be done with the use of a *Depth-First search* algorithm [CLRS09]. The Depth-First algorithm runs linearly in the size of the tree. The solving time is therefore exactly proportional to the tree size, or the number of paths.

If there is one access road in the middle of the road, then there is an exponential number of feasible schedules. Whenever $|\mathcal{I}|$ is large, the enumeration of the feasible schedules is not tractable and one should solve a single MILP instance of the form in (2.1) instead.

If there are two access roads, one on either side of the road, then there is an exponential number of feasible schedules. Again, if $|\mathcal{I}|$ is large, one should solve the MILP in (2.1).

Chapter 4

Algorithms

> An algorithm must be seen to be believed.
>
> Donald Knuth (1938–)

Model (2.1) is sufficient to solve the earth-moving problem. In this section we propose algorithmic techniques that improve the speed of convergence. In order to simplify the notation, we formulate (2.1) in standard form. Let m be the number of constraints in (2.1). Let n_c denote the number of continuous variables and $n = n_c + n_b$ denote the total number of variables in (2.1). We rewrite the inequality constraints and group the left-hand sides in a matrix $A \in \{0, \pm 1\}^{m \times n}$, and the right-hand side in a vector $b \in \mathbb{R}^m$, such that we can write (2.1) in the *canonical form*

$$\begin{array}{rl} \min & c^T x \\ \text{s.t.} & A(x,y)^T \geq b \\ & x \geq 0 \\ & y \in \{0,1\}^{n_b}, \end{array} \tag{4.1}$$

where $x \in \mathbb{R}^{n_c}$, $c \in \mathbb{R}^{n_c}_{++}$ is the cost vector, and $x \geq 0$ means component wise $x_i \geq 0$, for all $i \in \{1, \ldots, n_c\}$. We can solve (4.1) directly as a MILP, using a MILP solver software [Fou05]. Although this is not an algorithm in the common sense, we call it here the *direct solve* algorithm for future comparison.

Algorithm 1: Direct solve

```
Input: $x$,$y$,$c$,$A$,$b$ (from (2.1))
Output: Global minimum $z^* = c^T x^*$ for
        (4.1)
begin
    Build and solve MILP (4.1) for $x^*$;
end
```

4.1 Mass schedule as starting point

Definition 4.1. The LP obtained by relaxing the constraints $y \in \{0, 1\}^{n_b}$ with $y \in [0, 1]^{n_b}$, is called the *LP relaxation* of the MILP (4.1).

Fact 4.2 (Dantzig [Dan98, p. 516])**.** *The LP relaxation is a lower bound on the MILP.*

Most modern MILP solvers use a *branch-and-cut* algorithm [Fou05]. Branch-and-cut is a combination of the *branch-and-bound* method with the *cutting-plane* algorithm of Gomory [Gom58]. A classical *branch-and-bound* algorithm solves first the LP relaxation to the MILP. The algorithm then partitions the LP relaxation into subproblems which have one of the previously relaxed integer variables fixed to closest integer lower- and upper-bounds. The continuation of this process results in a branch-and-bound tree with nodes that represent subproblems. The bounding part of the algorithm looks for the best complete solution z^m in the tree. At each node k, where the solution z^k to the subproblem is greater than z^m, the algorithm can discard any further subproblem created by this node, since all following solutions will be greater than z^m. For a more detailed description of the branch-and-bound algorithm, we refer to [PS98].

A common technique to accelerate the solution process of a MILP is to provide an initial upper bound z° on the solution value z^m. This allows for early bounds on the branch-and-bound tree. An even better technique is to provide the solver with a feasible starting point (x°, y°), where $z^\circ = c^T x^\circ$. The number of nodes that need to be expanded can be reduced significantly with a starting point (x°, y°) that produces a sufficiently low upper bound z°. In this section, we describe a way to find a reasonable starting point.

In a traditional mass diagram procedure, work is performed in a unidirectional way from the start of the road to the end (see Remark 1.2). For a road with access attached to the starting section, a feasible block removal schedule is obtained by removing the blocks sequentially from the start of the road to the end, one block at each time step. If there is no access at the starting section of the road, we select the section with an attached access that is closest to the starting section and first remove sequentially all the blocks on the left of the access, and then all the blocks on the right. Once we have the schedule, we can fix the binary variables in (4.1) and solve the problem as a LP. This will give us feasible values for the continuous variables, and together with the fixed binary variables, a feasible solution to (4.1).

```
Procedure FindFeasiblePoint($x$,$y$,$c$,$A$,$b$)
  Input: $x$,$y$,$c$,$A$,$b$ (from (2.1))
  Output: Feasible point $(x^*, y^*)$ for (4.1)
  begin
      $a \leftarrow$ FindFirstAccess($\mathcal{S}$);
      for $k \leftarrow a$ to 1 do
          $y^*_{k,a-k+1} \leftarrow 1$;
      end
      for $k \leftarrow a+1$ to $|\mathcal{I}|$ do
          $y^*_{kk} \leftarrow 1$;
      end
      for $k \leftarrow 1$ to $|\mathcal{I}|$ do
          for $t \leftarrow 1$ to $|\mathcal{I}|$ do
              if $y^*_{k,t-1} = 1$ then $y^*_{kt} \leftarrow 1$;
          end
      end
      Solve (4.1) with fixed $y^*_{kk}$ as LP;
  end
```

We can now extend Algorithm 1 with procedure FindFeasiblePoint. We call this the *direct with* x° algorithm.

```
Algorithm 2: Direct with $x^\circ$
  Input: $x$,$y$,$c$,$A$,$b$ (from (2.1))
  Output: Solution $z^* = c^T x^*$ for (4.1)
  begin
      Build MILP (4.1);
      $(x^\circ, y^\circ) \leftarrow$FindFeasiblePoint($x$,$y$,$c$,$A$,$b$);
      Solve MILP (4.1) with $(x^\circ, y^\circ)$ for $(x^*, y^*)$;
  end
```

4.2 Maximal allowed moving distance $d_{\max}$

To reduce the model size, and hence reduce the solving time of the MILP, we introduce a heuristic that eliminates a significant number of variables. In order to do this, we introduce the parameter $d_{\max}$.

Definition 4.3. The *Maximal Allowed Moving Distance*, $d_{\max} \in \mathcal{S}$, is the largest number of sections over which earth is allowed to be moved. Therefore, for all $i, j \in \mathcal{M}, t \in \mathcal{T}$,

$$x_{ijt} = 0, \quad \text{if } |i - j| > d_{\max}. \tag{4.2}$$

The motivation behind $d_{\max}$ is that from practical experience, it makes little sense to move earth from a supply at one end of the road to a demand at the other end of the road. Intuition tells us that if there are other supplies along the road in between the two sections, then the closest supply to the demanding section should be chosen. Hence if we disallow moves that go over too long a distance, then we restrict the search space to shorter, and thus probably cheaper moves. We enforce $d_{\max}$ with a set of constraints in the form of Equation (4.2). If there are p such constraints, we can group them in a matrix $R_{d_{\max}} \in \{0,1\}^{p \times n}$ and reformulate the MILP (4.1) in a $d_{\max}$-extended form

$$\begin{array}{rl} \min & c^T x \\ \text{s.t.} & A(x,y)^T \geq b \\ & R_{d_{\max}} x = 0 \\ & x \geq 0 \\ & y \in \{0,1\}^{n_b}. \end{array} \tag{4.3}$$

Remark 4.4. It is clear from (4.2) that if $d_{\max}$ is small, then $R_{d_{\max}}$ becomes large (although very sparse). So instead of reducing the model size, we increase it. However, with each constraint in $R_{d_{\max}}$, we can remove a large number of feasible move constraints in A. This would be done before solving the model; potentially while building the model. Therefore, the model size is actually decreased. In fact, not only is the model size reduced, but also the problem becomes easier to solve due to the sparsity of $R_{d_{\max}}$.

Lemma 4.5. *For any optimal solution value $z^* = c^T x^*$ to the MILP* (4.1) *and any optimal solution value $z^*_{d_{\max}} = c^T x^*_{d_{\max}}$ to the extended form* (4.3)*, the inequality*

$$z^* \leq z^*_{d_{\max}}$$

is always true.

Proof. Let $\mathcal{C} = \{(x,y) : A(x,y)^T \geq b, x \geq 0, y \in \{0,1\}^{n_b}\}$ and $\mathcal{C}_{d_{\max}} = \{(x,y) : A(x,y)^T \geq b, R_{d_{\max}} x = 0, x \geq 0, y \in \{0,1\}^{n_b}\}$. Clearly, $\mathcal{C}_{d_{\max}} \subseteq \mathcal{C}$. It follows that $z^* \leq z^*_{d_{\max}}$. □

From Lemma 4.5 we can see that the optimality of the solution with respect to (4.1) is not guaranteed when using $d_{\max}$, unless $d_{\max} \geq |\mathcal{S}|$. For this reason, we call (4.3) the $d_{\max}$-*heuristic* to the original problem (4.1). Note that the demands and supplies for a particular vertical alignment are approximations. As such, a solution to (4.3) that is not optimal with respect to the original problem, may still be a good enough solution for an earth-moving problem. We have therefore an approximation algorithm that we call *direct with* x° *and* $d_{\max}$.

```
Algorithm 3: Direct with $x^\circ$ and $d_{\max}$
  Input: $x$,$y$,$c$,$A$,$b$ (from (2.1))
  Output: Solution $z^* = c^T x^*$ for (4.3)
  begin
      Build $R_{d^0_{\max}}$ using (4.2);
      $(x^\circ, y^\circ)$ ←FindFeasiblePoint($x$,$y$,$c$,$A$,$b$,$R_{d^0_{\max}}$);
      Solve (4.3) for $(x^0, y^0)$ with starting point
      $(x^\circ, y^\circ)$;
  end
```

Note that in this algorithm, we modified FindFeasiblePoint in order to take $R_{d^0_{\max}}$ into account. The change is implicit and we do not show the adapted procedure here.

4.3 Warmstarting with increasing $d_{\max}$

In Section 4.2, we mention that the $d_{\max}$-heuristic does not guarantee an optimal solution for (4.1). However, the solution from (4.3) can be used as a starting point to solve (4.1) using a warmstart procedure. Further improvement can be made by starting to solve (4.3) with a small $d_{\max}$ and use the solution to warmstart (4.3) with $\alpha d_{\max}$ where $\alpha \in \mathbb{R}_{++}, \alpha > 1$. If we continue this procedure, we end up with the following algorithm, which we refer to as *warmstart* algorithm.

```
Algorithm 4: Warmstart
  Input: $x$,$y$,$c$,$A$,$b$,$d^0_{\max} \geq 1$,$\alpha > 1$
  Output: Solution $z^* = c^T x^*$ for (4.1)
  begin
      $k \leftarrow 0$;
      Build $R_{d^0_{\max}}$ using (4.2);
      $(x^\circ, y^\circ) \leftarrow$FindFeasiblePoint($x$,$y$,$c$,$A$,$b$,$R_{d^0_{\max}}$);
      Solve (4.3) for $(x^0, y^0)$ with starting point
      $(x^\circ, y^\circ)$;
      while $d^k_{\max} < m_s$ do
          $k \leftarrow k + 1$;
          $d^k_{\max} \leftarrow \alpha d^{k-1}_{\max}$;
          Build $R_{d^k_{\max}}$ using (4.2);
          Solve (4.3) for $(x^k, y^k)$ with starting point
          $(x^{k-1}, y^{k-1})$;
      end
      $(x^*, y^*) \leftarrow (x^k, y^k)$
  end
```

Lemma 4.6. *The warmstart method in Algorithm 4 converges to the optimal solution value z^* of the MILP* (4.1)*.*

Proof. Recall that m_s is the number of sections. Let $1 \leq d^0_{\max} < m_s$ and $\alpha > 1$. Hence there exists $k \in \mathbb{Z}_+$, such that $\alpha^k d^0_{\max} \geq m_s$. Therefore, at iteration k of Algorithm 4, we have

$$d^k_{\max} = \alpha d^{k-1}_{\max} = \alpha\alpha \cdots \alpha d^0_{\max} = \alpha^k d^0_{\max} \geq m_s.$$

For this k, the extended MILP (4.3) is equal to the original MILP (4.1). □

4.4 Warmstarting with mipgap check

Definition 4.7. The *mipgap* $|z^k - z^*_{\text{LP}}|$ is the gap between the current best feasible value of the MILP, z^k, and the optimal value of its LP relaxation, z^*_{LP}.

Lemma 4.8. *The optimal value z^*_{LP} of the LP relaxation for to the original MILP* (4.1) *gives a lower bound for the $d_{\max}$-extended MILP* (4.3)*.*

Proof. Combine Fact 4.2 and Lemma 4.5. □

Lemma 4.9. *Let* z^*_{LP} *be the optimal value for the LP relaxation of* (4.1) *and* z^* *be the optimal value for the MILP* (4.1)*. For any current best feasible value* $z^k_{d_{\max}}$ *of* (4.3)*,*

$$|z^* - z^*_{\mathrm{LP}}| \leq |z^k_{d_{\max}} - z^*_{\mathrm{LP}}|.$$

Proof. The result follows from Lemma 4.5 and Lemma 4.8. □

We can now extend Algorithm 4 with a stopping condition that verifies the mipgap. We call this the *warmstart with* ϵ algorithm.

```
Algorithm 5: Warmstart with $\epsilon$
Input: $x$,$y$,$c$,$A$,$b$,$d^0_{\max} \geq 1$,$\alpha > 1$,$\epsilon > 0$
Output: Solution $z^* = c^T x^*$ for (4.1), within $\epsilon$
begin
    Solve for the LP relaxation solution $z^*_{\mathrm{LP}}$ of (4.1);
    $k \leftarrow 0$;
    Build $R_{d^0_{\max}}$ using (4.2);
    $(x^\circ, y^\circ) \leftarrow$FindFeasiblePoint($x$,$y$,$c$,$A$,$b$,$R_{d^0_{\max}}$);
    Solve (4.3) for $z^0 = c^T x^0$ using starting point $(x^\circ, y^\circ)$;
    while $d^k_{\max} < m_s$ and $\epsilon < |z^k - z^*_{\mathrm{LP}}|$ do
        $k \leftarrow k + 1$;
        $d^k_{\max} \leftarrow \alpha d^{k-1}_{\max}$;
        Build $R_{d^k_{\max}}$ using (4.2);
        Solve (4.3) for $(x^k, y^k)$ with starting point $(x^{k-1}, y^{k-1})$;
    end
    $(x^*, y^*) \leftarrow (x^k, y^k)$
end
```

In the next chapter, we look at the numerical results for Algorithms 1 - 5 with a set of problems based on real road alignment designs. We analyze solving times and quality of solutions for the different algorithms and try to find good values for the parameters α and $d_{\max}$.

Chapter 5

Numerical results

Understanding is, after all, what science is all about-and science is a great deal more than mere mindless computation.

Roger Penrose (1931–)

There are two categories of questions that we seek to statistically examine in this section. First, which algorithm is the fastest, and which algorithm is the most robust to solve our MILP model? Second, we want to know how the solution value of our model compares to the solution value of a LP solution for a given non-optimal removal schedule.

5.1 Experimental setup

5.1.1 Problems

In order to compare the algorithms in Chapter 4, we based our test problems on a family of real road alignments. We used a road design software to generate 8 different road alignments with length between 1 and 40 kilometers. We denote the alignments with letters from A to H. From the 8 designs, we then created 44 problems with different block and access road configurations. For all of the problems we added two access roads, one at the beginning and one at the end of the road. For about half of all the problems we added two additional access roads in between the start and the end of the road. We further split each group into three subgroups of 5, 10, and 20 blocks. The problem configurations are given in Table A.1 of Appendix A.

5.1.2 Benchmark

In order to evaluate the benefits of our model over current practices, we would need to conduct a field study on a sufficiently large set of real construction projects. Similar to our theoretical problem set, such projects would need to have obstacles and access roads in different configurations. The effective earthwork costs could then be compared with the solution of our model. In our research, we did not have access to such data. We therefore tried to build a reference solution that, to our knowledge, would be as fair as possible to compare with our solution.

Reference solution

For our reference solution we assume that obstacles are removed in a sequential order. We assume that earthwork tasks can occur at the same time as the obstacle removal process, as long as the sections for these tasks are accessible. We therefore use the removal scheme in Procedure FindFeasiblePoint as reference block removal schedule. As in Procedure FindFeasiblePoint, we solve the LP for the fixed schedule. Note that we do not use any $d_{\max}$ restrictions in the reference solution. We call this reference solution the *LP solution for mass removal schedule.*

We are aware that block removal schedules vary for different construction projects and they also depend heavily on the experience of the geotechnical engineer. A mass removal schedule, however, is perfectly valid in forest areas or in ground conditions where the operation of ordinary construction equipment is not possible. In very difficult ground settings, tractor mounted equipment is used to push large-sized granular material in front of the equipment to build a stable construction platform [Fwa06]. This corresponds to a sequential removal of ground obstacles.

Parameters

For some of the tests we changed parameters for the algorithms in Chapter 4, such as:

- ε is the tolerance for the MILP solver;
- ε_{LP} is the LP optimality tolerance to solve the LP relaxation;
- ϵ is the mipgap defined in Definition 4.7;
- $d_{\max}$] is the maximal allowed moving distance;
- α is the scaling factor for $d_{\max}$ within the warmstarting scheme.

Note that except for the warmstart algorithms, $\varepsilon = \epsilon$. For the warmstart algorithms, we use

$$\varepsilon = \begin{cases} 0.05 & \text{if } d_{\max}^k < m_s, \\ \epsilon & \text{if } d_{\max}^k \geq m_s. \end{cases}$$

In some problems, the solving time for the LP relaxation z_{LP}^* (see Algorithm 5) can be quite long. To reduce this solving time, we change the LP solver optimality tolerance $\varepsilon_{\mathrm{LP}}$. The LP solver in the Gurobi Optimizer has a default LP optimality tolerance of $\varepsilon_{\mathrm{LP}} = 10^{-6}$. An increase in the LP optimality tolerance results in a faster solving time for an LP. Now if for the mipgap tolerance $\epsilon > 0$, we have

$$|z^* - z_{\mathrm{LP}}^*| \leq \epsilon,$$

then for some $\delta > 0$, by the triangle inequality, we have

$$|z^* - z_{\mathrm{LP}}^* + \delta| \leq \epsilon + \delta.$$

So if we let the new LP optimality tolerance $\tilde{\varepsilon}_{\mathrm{LP}} > \varepsilon_{\mathrm{LP}}$, with $\delta = \tilde{\varepsilon}_{\mathrm{LP}} - \varepsilon_{\mathrm{LP}}$ and the new mipgap tolerance $\tilde{\epsilon} = \epsilon + \delta$, then the convergence of Algorithm 5 within a mipgap tolerance of ϵ is still guaranteed. For our experiments with the warmstart and ϵ ($d_{\max}^0 = 25, \alpha = 10$) and warmstart with ϵ ($d_{\max}^0 = 5, \alpha = 5$) algorithm, we used $\tilde{\varepsilon}_{\mathrm{LP}} = 10^{-5}$ to solve the LP relaxation and $\tilde{\epsilon} = \epsilon + \delta$ for the mipgap tolerance. For all the other computations, we used the default LP optimality tolerance.

Test configurations

In our first set of experiments, our goal is to investigate the behavior of the solution and solving time with respect to $d_{\max}$. Hence we run all the problems with the direct with x° and $d_{\max}$ algorithm (Algorithm 3). The configuration for these tests are given in Table 5.1. In our second set of experiments, our goal is to investigate the economical benefit and the performance of the algorithms in Chapter 4. The LP solution for mass removal schedule was used for an economical comparison to the exact solution to (2.1), as given by the algorithms in Chapter 4. Furthermore, a performance analysis was done. The LP with mass removal schedule was not used in the performance comparison since it is a pure LP, and it is of little value to compare the solving time of a single LP with the algorithms in Chapter 4. Moreover, the LP with mass removal schedule is not an optimal solution to (2.1). We therefore focus our performance analysis on the MILP

Table 5.1: Test configurations for $d_{\max}$ with the direct solve algorithm using x° (Algorithm 3)

ϵ	$d_{\max}$	ϵ	$d_{\max}$
0.01	5	0.05	5
	25		25
	100		100
	200		200

Table 5.2: Economical test and performance test configurations

ϵ	$d^0_{\max}$	α	Algorithm
0.01	-	-	direct solve (Algorithm 1)
			direct with x° (Algorithm 2)
	25	10	warmstart (Algorithm 4)
			warmstart with mipgap (Algorithm 5)
	5	5	warmstart with mipgap (Algorithm 5)
0.05	-	-	direct solve (Algorithm 1)
			direct with x° (Algorithm 2)
	25	10	warmstart (Algorithm 4)
			warmstart with mipgap (Algorithm 5)
	5	5	warmstart with mipgap (Algorithm 5)

algorithms only. In the context of economical analysis and performance, we run the experiments with the configurations shown in Table 5.2. For all our tests, we used a CPU time limit of 12 hours. If an algorithm could not solve the problem within the time limit, we aborted the solving process and considered the problem as not solved.

Remark 5.1. In practice, a MILP tolerance of 5% is sufficiently precise for an earth-moving problem. The data for earth quantities is based on approximations that mostly use triangular surface grids. The use of triangularization, measurement errors, and non-constant factors for earth shrinkage and swelling, introduce errors that are in the order of 5%.

Test equipment and software

All the experiments were solved with the Gurobi Optimizer 2.0.2 MILP solver (`http://www.gurobi.com`) on a standard 64-bit Linux (Ubuntu 9.10 Server) workstation with an Intel® 2.4GHz Quad-Core CPU and 8GB of memory. Since the solver makes use of multiple cores, the reported CPU-time may be greater than the wall-clock time in some problems. The algorithms were implemented in C++ and the code can be made available upon request. Please contact the author at valentin.koch@gmail.com.

5.2 Results

5.2.1 Analysis of the $d_{\max}$-heuristic

In the top row in Figure 5.1 we show the boxplots of the relative error of the objective value with respect to the solution for $d_{\max} \geq m_s$ and the corresponding MILP solver tolerances of ε=1% and ε=5%. The bottom row contains the relative solving time with respect to the solving time for $d_{\max} \geq m_s$. In the boxplots, the box represents the *interquartile range* (IQR) and contains 50% of the sample points. It gives therefore a relatively robust statistic. The vertical bar inside the box represents the median. The ends of the whiskers represent the lowest datum still within 1.5 IQR of the lower quartile, and the highest datum still within 1.5 IQR of the upper quartile. Outliers are depicted as small dots. For convenience, the mean is also plotted as a small cross. We only counted problems that could be solved for both, *varepsilon*=1% and ε=5%. The numbers for the plots are given in Table A.2 and A.3 in the Appendix.

Looking at $d_{\max} = 5$ in the 1% configuration, we see that the median relative error of 34.84% for the objective value is about 5% smaller than the

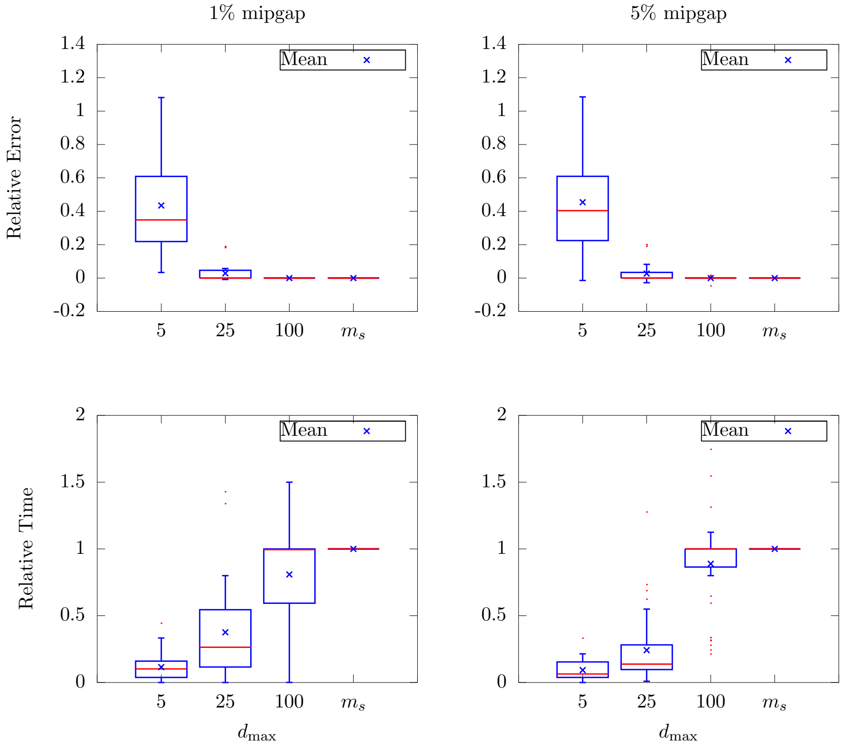


Figure 5.1: Objective and solving time with increasing $d_{\max}$ and different tolerance

median relative error of 40.39% for the 5% configuration. Remember that the reference solution for the relative error is also solved within a tolerance of 1% or 5%, respectively. With regard to Remark 5.1, we conclude that, when using $d_{\max} = 5$, the relative error is too large for a solution to be acceptable. At $d_{\max} = 25$, however, the median error is 0% in both cases, with an IQR of 4.6% and 3.3%, respectively. At the same time, we can look at the relative solving time and we can see that for 1% tolerance, at $d_{\max} = 5$, we need 10.2% of the time that the direct solve with x° algorithm uses to solve the MILP. The time for a 5% tolerance at $d_{\max} = 5$ is slightly smaller, at 6.4% of the direct solve and x°, which can be explained with the earlier convergence. For $d_{\max} = 25$ we ended up with 26.4% and 13.8%, respectively. A quick glance at the figure tells us that there is not much difference at $d_{\max} = 100$ to the direct solve algorithm. Consider, however, that 80% of all problems have $m_s \leq 100$ and therefore have the same results with $d_{\max} = 100$ and $d_{\max} \geq m_s$.

If we add the IQR to the solving tolerance, we see that a choice of $d_{\max} = 25$ produces a reasonable result that lies usually within 5.6% of the real solution with 1% solving tolerance, and within 4.3% of the solution with 5% solving tolerance. The results with 1% solving tolerance do not lie within the 5% suggested by Remark 5.1. However, a solution from $d_{\max} = 25$ is sufficiently close to the real solution in order to give a good upper bound for a subsequent MILP with a larger $d_{\max}$ or $d_{\max} \geq m_s$.

The median solving time with $d_{\max} = 5$ for 1% tolerance is around 10 times faster, and with 1% tolerance around 16 times faster than the direct with x° algorithm. In more than half of all the problems, the solving time does not exceed 16% of the solving time with the direct and x° algorithm for both tolerances. In the $d_{\max} = 25$ case, the solving time is about 4 and 7 times faster. We observe that the upper quartile for 1% is at about half and for 5% at about a third of the solving time with the direct with x° algorithm. We will use these observations in the analysis of the warmstart scheme in order to derive the effect of the starting point given by $d_{\max} = 5$ and $d_{\max} = 25$.

5.2.2 Performance

In order to compare the performance of the algorithms, we use a *performance profile* as developed in [DM02]. For each algorithm, we plot the number of problems that are solved within a factor of the best CPU time. Let $\mathcal{P}$ be the set of all problems, and $\mathcal{A}$ be the set of algorithms. We compute the *performance ratio* by

$$r_{p,a} = \frac{t_{p,a}}{\min\{t_{p,a} : a \in \mathcal{A}\}},$$

where $p \in \mathcal{P}$, and $t_{p,a}$ is the solving time of algorithm a for problem p. The percentage of problems that are solved within a factor $\tau \in \mathbb{R}$ are computed by

$$\rho_a(\tau) = \frac{|\{p \in \mathcal{P} : r_{p,a} \leq \tau\}|}{|\mathcal{P}|}.$$

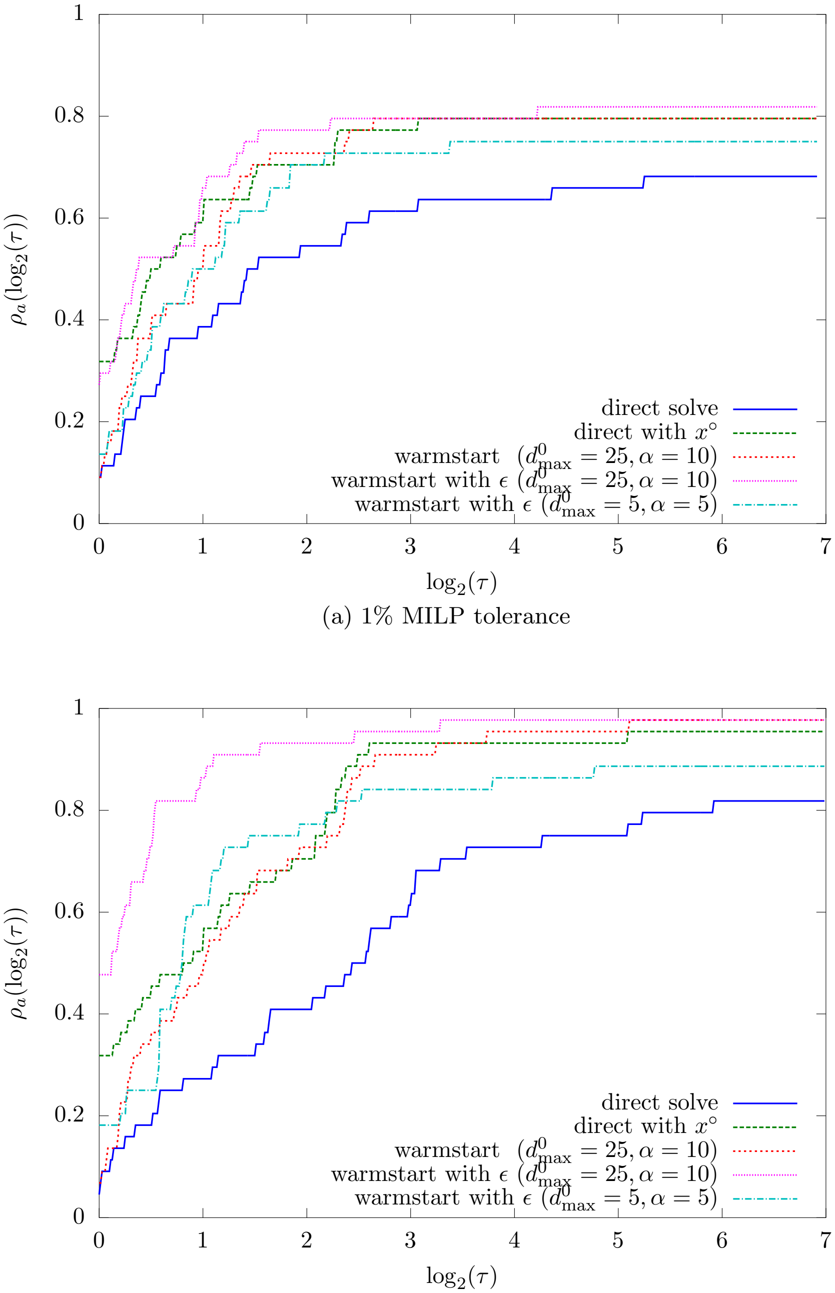
Hence for a particular problem p, an algorithm a needs τ times longer than the best algorithm for p, if $\tau > 1$ when algorithm a is able to solve it. If algorithm a needs only $\tau = 1$, the algorithm is considered as the best, or among the best algorithms for problem p. The percentage ρ_a of all problems solved by algorithm a at τ gives us an overall assessment of the performance of algorithm a. We conclude that the algorithm with the highest ρ value is likely to perform best on a given problem with a probability of $\rho_a(\tau)$. For a good algorithm, the plot of $\rho(\tau)$ should therefore be found above the other algorithms. If we increase τ to allow the algorithms to be within a larger factor of the best algorithm, then the algorithm with the highest ρ is the most robust algorithm. Hence an algorithm with the highest ρ value at a large τ, solves most of the problems. For a more detailed description of performance profiles, we refer the reader to [DM02].

Figure 5.2 shows the performance profiles for Algorithms 1 to 5 on a $\log_2$ scale, that is,

$$\rho_a(\tau) = \frac{|\{p \in \mathcal{P} : \log_2(r_{p,a}) \leq \tau\}|}{|\mathcal{P}|}.$$

Figure 5.2a shows the profile for ε=1% and Figure 5.2b shows the profiles with ε=5%. In Figure 5.3a and 5.3b, the profiles are shown on a normal scale with a short τ interval. Figures 5.4a and 5.4b show the profiles on a normal scale with a long τ interval. For all profiles, we have $|\mathcal{A}| = 5$ and $|\mathcal{P}| = 44$.

For a ε=1%, we can see that the direct solve with x° algorithm outperforms the other algorithms with a probability of about 32%, and the probability that the warmstart with ϵ ($d^0_{\max} = 25, \alpha = 10$) algorithm outperforms all other algorithms is at about 27%. If we increase τ to let the algorithms be within a factor of 4 of the direct solve with x° algorithm, we can see that the warmstart with ϵ ($d^0_{\max} = 25, \alpha = 10$) algorithm wins with a high probability of about 78% and solves up to 80% of the problems when τ is extended to 5. We can see that the warmstart with ϵ ($d^0_{\max} = 25, \alpha = 10$) is the most robust



(a) 1% MILP tolerance

(b) 5% MILP tolerance

Figure 5.2: Algorithms performance profiles with $\log_2$ scale

algorithm (82% of all problems solved), followed by the warmstart with not mipgap check and the direct solve with x° algorithm (79.5% of all problems solved). We conclude that the warmstart with ϵ ($d^0_{\max} = 25, \alpha = 10$) algorithm is a good choice to solve most of the problems relatively fast. Clearly, the direct solve algorithm is not very competitive in these experiments.

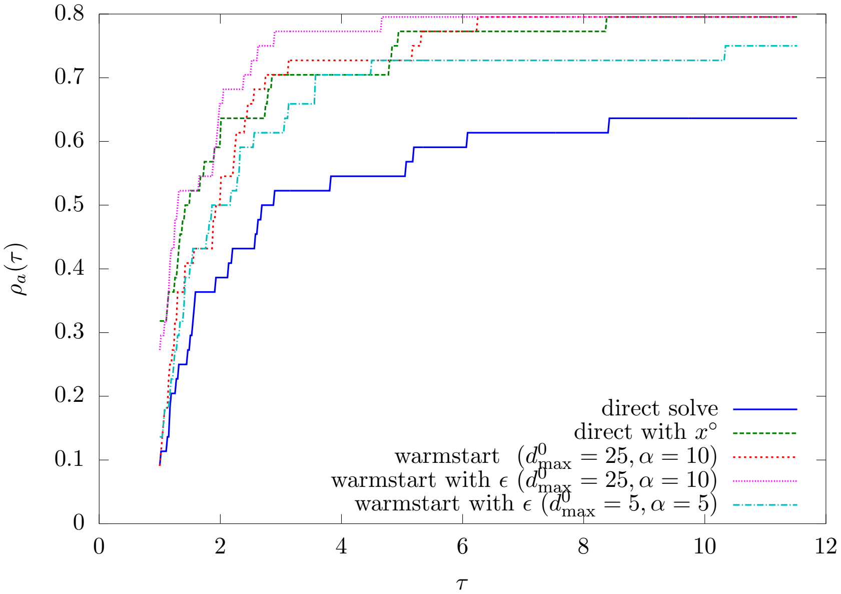
The warmstart with ϵ ($d^0_{\max} = 25, \alpha = 10$) converged with a $d_{\max} \leq 25$ on about 34% of all solved problems. Recall that in Section 5.2.1, for ε=1%, we found that the median and lower quartile of the relative solution error for $d_{\max} = 25$ is 0% with an upper quartile of 4.6%. Therefore half of the solutions were above 4.6% of the real solution. From Lemma 4.9 we expect that the warmstart with ϵ algorithm will use at least one more iteration for these solutions. Furthermore, due to the IQR of 4.6% for the solution error, there is also a significant number of solutions within the IQR that are above the required ε=1% tolerance. Altogether, these problems account for the 66% where the warmstart with ϵ algorithms were required to increment $d_{\max}$ from 25 to 125 or 250, depending on α.

For ε=5%, the warmstart with ϵ ($d^0_{\max} = 25, \alpha = 10$) algorithm dominates all other algorithms on 46% of all problems. The algorithm clearly stays ahead of the others until $\tau \approx 35$. At that point, either the warmstart with ϵ ($d^0_{\max} = 25, \alpha = 10$) or the warmstart without mipgap check would suffice to solve about 97.7% of all the problems. The warmstart with ϵ ($d^0_{\max} = 25, \alpha = 10$) algorithm is therefore the clear winner in the ε=5% experiments.

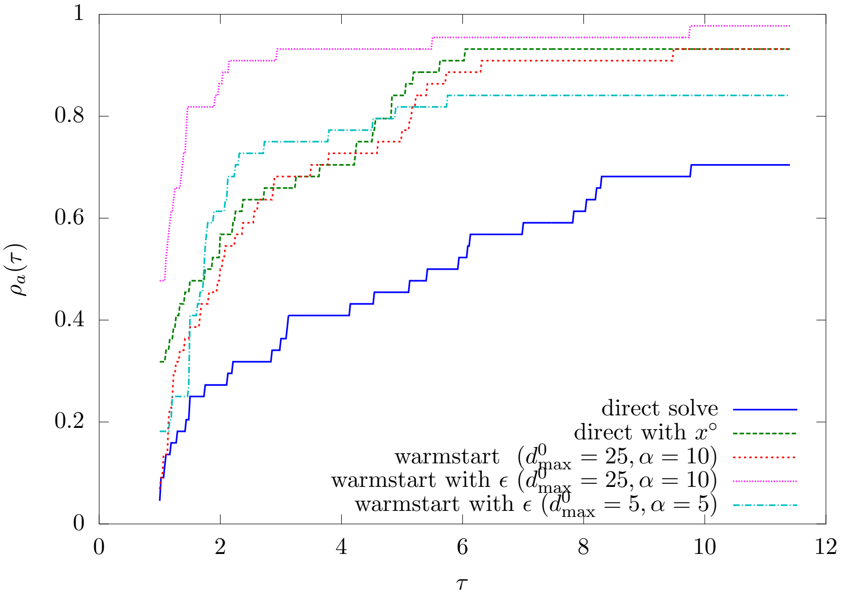
An explanation why the warmstart with ϵ ($d^0_{\max} = 25, \alpha = 10$) algorithm performs always better or equal to the other algorithms when ε=5% lies in the mipgap check. While the median and lower quartile for the solution error at $d_{\max} = 25$ are, similar to the ε=1% experiments, at 0%, the upper quartile is only at 3.3%. Therefore half of all the problems that lie in the IQR at $d_{\max} = 25$ are already within ε=5%. From the results, we know that the algorithm converged on 61% of all solved problems with a $d_{\max} \leq 25$. This is a notable result that confirms our $d_{\max}$-analysis.

The warmstart with ϵ ($d^0_{\max} = 5, \alpha = 5$) algorithm shows a mixed performance. It starts better than the simple warmstart with no mipgap check algorithm and if $\tau \approx 3$, it also outperforms the direct with x° algorithm with a probability of over 70%. However, after $\tau = 6$, the warmstart with ϵ ($d^0_{\max} = 5, \alpha = 5$) algorithm performs only better than the direct solve algorithm and finished by solving 88.6% of all the problems. The direct solve algorithm solved only 81%.

In the next section, we will analyze the quality of the solutions with respect to this LP with mass removal schedule.

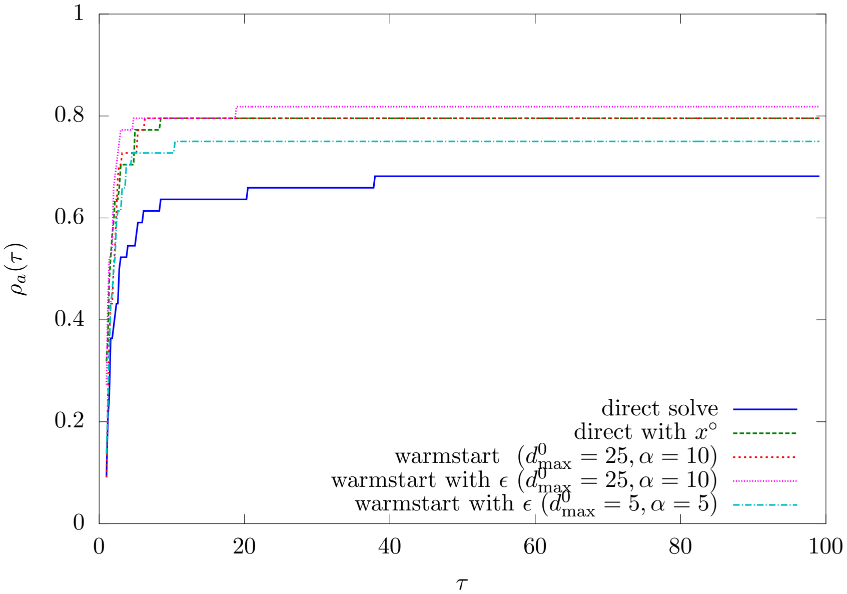


(a) 1% MILP tolerance

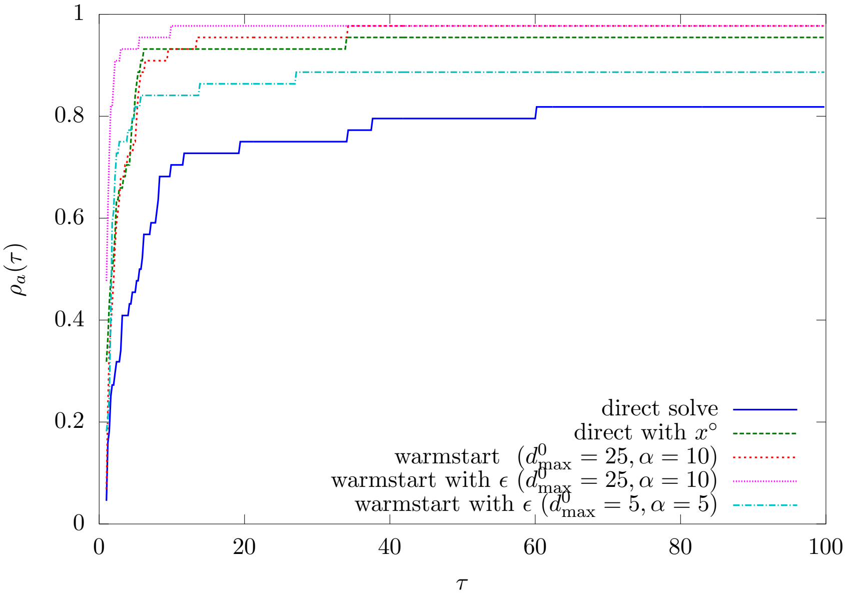


(b) 5% MILP tolerance

Figure 5.3: Algorithms performance profiles with normal scale and small intervale for τ



(a) 1% MILP tolerance



(b) 5% MILP tolerance

Figure 5.4: Algorithms performance profiles with normal scale and large interval for τ

5.2.3 Economical analysis

In Section 5.1.2, we defined the LP with mass removal schedule as our reference solution. Our reference solution is based on a LP model and does not necessarily represent the real cost of an actual earthwork construction project with current techniques.

Let $z^*_{p,\mathrm{ref}}$ be the solution for problem p using the LP with mass removal schedule. Let $z^*_{p,a}$ be the solution for problem p using algorithm $a \in \mathcal{A}$. We define the gain $g_{p,a}$ by

$$g_{p,a} = 1 - \frac{z^*_{p,a}}{z^*_{p,\mathrm{ref}}}.$$

For all problems that have been solved, we plot the gain in a histogram. The number of bars in the histogram is the square root of the number of problems that have been solved. We also plot a normal curve over the histogram that fits the data. Note that we observed that in some instances where the problems were not solved, the current solution was still better than the solution of the LP with mass removal schedule. It is not clear if one should count these solutions in the histogram plots. We decided to keep these solutions in our economical analysis on the following grounds. In practice, if a solver is interrupted with an intermediate solution that is better than a traditional LP solution, the intermediate solution will most likely be used.

In Section 5.2.2 we identified the best three performing algorithms as the warmstart with ϵ ($d^0_{\max} = 25, \alpha = 10$) algorithm (Algorithm 5), the direct with x° algorithm (Algorithm 2), and the warmstart ($d^0_{\max} = 25, \alpha = 10$) algorithm (Algorithm 4). Since the warmstart with ϵ ($d^0_{\max} = 25, \alpha = 10$) algorithm solved more problems than the other two algorithms, we focus our economical analysis on the results of the warmstart with ϵ ($d^0_{\max} = 25, \alpha = 10$) algorithm. Figure 5.5 shows the histogram for the warmstart with ϵ ($d^0_{\max} = 25, \alpha = 10$) algorithm. Histograms for the other two algorithms are shown in Appendix B. The statistics for the histograms can be found in Appendix A, Tables A.4 and A.5.

For the problems that have been solved, the use of the warmstart with ϵ ($d^0_{\max} = 25, \alpha = 10$) algorithm and ε=1% produced an average gain of 4.1%. The same algorithm delivered an average of 3.3% gain if ε=5%. From the plot we see that the negative gains in the ε=5% histogram, that are caused by the MILP tolerance, have a much greater influence on the average than in the ε=1% case, where almost no negative gains are encountered. The maximum gains are 17.9% in the ε=1% histogram, and 19% in the ε=5% histogram.

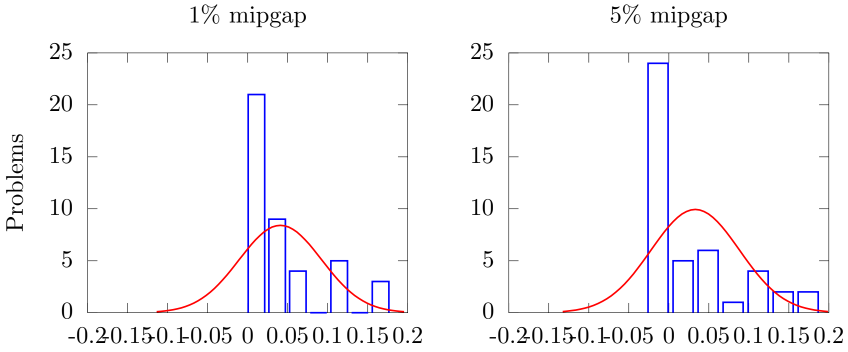


Figure 5.5: Savings with warmstart with ϵ ($d^0_{\max} = 25, \alpha = 10$), compared to mass block removal schedule

Overall we can observe that the average gains for ε=1% are not significantly higher than for ε=5%. This is surprising since in general, the MILP solver stops improving the solution in the ε=5% case much earlier than in the ε=1% case. From the tables in the appendix we can see that the solving time for the ε=1% case is much higher than for ε=5%. The qualitative merit of our model can be assessed when looking at the best gains. For each of the two MILP tolerances, there were 8 problems with gains of 10% and more. This accounts for about 18% of all the problems. In the context of a large highway project, this is a significant amount of gain over the LP with mass schedule.

We are interested in the problems that produced the biggest gains. In order to see if there are correlations between problem types and gains, we did a stratification analysis with the use of a scatter plot. The scatter plot is shown in Figure 5.6. The scatter plots for the other algorithms can be found in Appendix B. We first grouped the problems into the number of blocks. Each group is shown with a different color and symbol. Green stars show five block problems, blue circles show ten block problems, and red triangles show twenty block problems. Each group was internally sorted by the number of sections and plotted from left to right in ascending order. Hence, problem 1 in the scatter plot represents the five block problem with the least number of sections, whereas problem 17 represents a five block problems with the largest number of sections. Small points represent problems with two access roads, large points represent problems with four access roads.

In the stratification analysis, we could not identify any visibly striking pattern. The ε=1% and the ε=5% results have a similar distribution. One might observe that in the groups of five block problems, there are less solu-

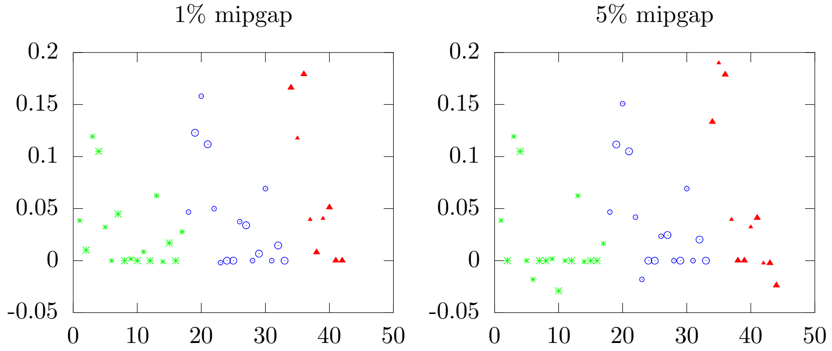


Figure 5.6: Stratification of gains for warmstart with ϵ ($d^0_{\max} = 25, \alpha = 10$); * are 5, $\odot$ are 10, ▲ are 20 blocks; small symbols are 2, big symbols are 4 access roads; within the block groups, points are sorted from left to right in ascending order of section numbers

tions with significant gains than in the other two groups. For these problems, it seems, the LP with mass removal schedule provides often a good enough solution. If we group the results with the highest gains, we can observe a majority of problems with four access roads. Intuition tells us that with more access roads, there is more flexibility and hence a potential for cheaper solutions. We conclude that for a better stratification analysis, experiments with a larger set of problems would certainly give more insight.

Chapter 6

Conclusion

Frustra fit per plura quod potest fieri per pauciora.[4]

William of Ockham (1287-1347)

6.1 Accomplishments

We started the thesis with the description of traditional methods to balance earthwork and minimize hauling costs in the context of road construction. Two main methods were presented in Chapter 1. First, we explained the mass diagram, a simple tool that helps to balance the earthwork and to obtain an estimate on the directions of haul. Second, we showed a more detailed model of a LP by Mayer and Stark [MS81] that not only balances the earthwork, but also minimizes the hauling cost. We then pointed out that neither method captures the problem of blocks that restrict some of the earth movements to balance earth across the construction site. To address this problem, we presented an extension to the model of [MS81] in Chapter 2. In Chapter 3, we analyzed the model with respect to space complexity and we investigated the growth of the number of feasible block removal schedules. To improve the computation time for solving the MILP, we outlined a set of algorithms in Chapter 4. In Chapter 5, we presented the numerical results of the algorithms, together with an analysis of performance, robustness, and accuracy.

6.2 Research outcomes

In the model analysis, we first identified a potential issue of memory growth. With a fixed number of blocks, we found that the theoretical upper bound on the growth of memory for the MILP model in (2.1) is $\mathcal{O}\left(|\mathcal{M}|^4\right)$. In

[4] It is pointless to do with more what can be done with less.

our experiments, we use problems with $|\mathcal{M}| \lesssim 200$. However, models with $|\mathcal{M}| \approx 200$ required relatively large amounts of the available memory. The solver failed to solve some of these problems due to memory exhaustion.

A second result of our model analysis was the enumeration of the feasible block removal schedules. We concluded that for a road with one access road at either end, there is no need to solve for the MILP, and one can just solve a LP for a fixed removal schedule that removes the blocks sequentially. If the access road is at some other place, we can quickly compute the number of feasible schedules with equation (3.4). If the number of feasible schedules is small, we can enumerate the schedules and solve a LP for every schedule. The schedule that produces the smallest objective value is the optimal removal schedule. We also showed that the worst case of number of schedules for one access road is bounded by $\mathcal{O}\left(3^{|\mathcal{I}|}\right)$. Therefore, if $|\mathcal{I}|$ is large, we recommend to solve the MILP instead of a schedule enumeration. The same observations were made for the typical configuration of two access roads, one on each end.

In our numerical analysis, we showed that the MILP can be solved efficiently with the help of the algorithms from Chapter 4. In the experiments, we demonstrated the strength of the $d_{\max}$-heuristic, and showed that there is a significant performance benefit when using the warmstart with ϵ, the simple warmstart, or the direct with x° algorithm, instead of solving the MILP directly. Furthermore, we showed that these three algorithms are considerably more robust than a method that solves directly for the MILP.

As a last observation in our analysis, we showed the economic benefits of solving the MILP instead of using a LP with a fixed mass removal schedule. If we solve a problem to an accuracy of ε=1%, the MILP produces solutions that are about 4.1% cheaper in the average. If we aim for ε=5% accuracy, the MILP provides us with a solution that saves roughly 3.3% on a LP solution for a fixed mass removal schedule. At a first sight, these benefits may not seem very high. However, in the context of a highway project that involves earthwork costs of millions of dollars, 4.1% is a significant figure.

In summary, we started by presenting a MILP model that overcomes the problems of the mass diagram and the LP model by Mayer and Stark. We pointed out potential issues with the model and efficient ways to solve the MILP. The result of the model presents an engineer with a good estimate on earthwork costs. The model is therefore a good tool to quantify a given vertical alignment, and therefore provides a valuable tool to find the most economic alignment.

As already pointed out by Mayer and Stark [MS81], the objective of an earthwork contractor is to maximize its profit. A contract that is based on an engineer's mass diagram, and that fixes the costs for excavation, borrow, and haul, may not reflect additional costs that result from a haul scheme that is not possible due to blocks. The presented MILP enables a contractor to estimate the real earthwork costs of a project, and at the same time provides the contractor with the cheapest haul scheme in order to maximize its profit.

6.3 Future research

6.3.1 Warmstart schemes and ε_{LP}

In Section 5.2.2, we explained the problems that we encountered when solving the LP relaxation with the default LP optimization tolerance of the Gurobi Optimizer. At the writing of this thesis, preliminary results of tests with $\varepsilon_{\text{LP}} = 10^{-5}$ suggest an even better performance for the warmstart with ϵ ($d^0_{\text{max}} = 25, \alpha = 10$) and warmstart with ϵ ($d^0_{\text{max}} = 5, \alpha = 5$) algorithms. We also would like to investigate additional warmstart schemes with different d^0_{max}- and α-values.

6.3.2 Vertical alignment

The above model can easily be extended to the vertical alignment. Based on a vertical optimization model by Easa [Eas87], Moreb introduced a LP model that combines the earthwork model by Mayer and Stark with roadway grade selection by using a piecewise linear approximation of the road profile [Mor96]. In [MA04], Moreb and Aljohani extended the model by using a quadratic approximation. The quadratic representation was then further extended in [Mor09] with a spline approximation, with improvements by Koch and Lucet [KL10]. The result was an improved and efficient linear model that avoids sharp connectivities and scaling problems of piecewise linear representations.

The main difference between the LP model of Mayer and Stark, and the vertical alignment model by Moreb, is the earth volume. In the earthwork LP and in our MILP model, we treat the volume of a section i as a fixed constant V_i. We recall the balance constraint (1.3b) from the LP as

$$\sum_{\substack{j \in \mathcal{M} \\ j \neq i}} (x_{ij} - x_{ji}) = V_i \qquad \text{for all } i \in \mathcal{S}.$$

However, in the vertical alignment LP by Moreb, the volume is a linear approximation. The volume is computed as the product of the surface area

A_i of section i, with the average height of earth that is cut from the section or filled into that section. The cut or fill is represented with LP decision variables, namely u_i for the height of the cut, and v_i for the height of the fill in that section. The balance constraint then becomes

$$\sum_{\substack{j \in \mathcal{M} \\ j \neq i}} (x_{ij} - x_{ji}) = A_i(u_i - v_i) \qquad \text{for all } i \in \mathcal{S}.$$

The ground profile can then be modeled with spline segments. The spline segments do not need any additional decision variables besides u_i and v_i. We refer the reader to [Mor09, KL10] for the details. Since the model for vertical alignment needs only two additional continuous decision variables per section, the MILP model in (2.1) can readily be combined with the LP model in [Mor09, KL10].

6.3.3 Schedule enumeration

In Remark 3.25, we discussed the use of feasible schedule enumeration in the case of a small number of blocks. The use of a Depth-first search algorithm allows an enumeration in linear time. However, the number of feasible schedules itself is exponential. A natural question is what a small number of blocks means in this context. It would be interesting to compare the solving time for the enumeration of schedules, together with solving the LP for each schedule, and the solving time for the MILP. We believe that up to a certain number of blocks, the enumeration of all LP solutions would be significantly faster than solving the MILP. The decision process between the two methods would be simplified significantly with the identification of a threshold that indicates the number of blocks where the MILP will outperform an exhaustive enumeration.

An additional topic of interest, with regard to the schedule enumeration, is the use of *dynamic programming*. In Section 3.2, we described a removal schedule in terms of binary strings, one binary string for each time step. An interesting observation is that the same binary string can be used in multiple schedules. In fact, after a certain time step, multiple schedules may even end with the same set of binary strings. We can therefore look at a binary string as a *state* in the removal process[5]. Once two schedules share the same state, they need to solve the same subproblem to finish the removal process. Dynamic programming is particularly well suited for optimization problems with overlapping subproblems [Bel52]. It would be interesting to formulate

[5] This observation was made by Jim Nastos.

the feasible schedule enumeration as a dynamic program, and to compare the method with exhaustive enumeration, as well as the MILP method. If a threshold was found for the exhaustive enumeration and the MILP, it would be interesting to see if the threshold changes with dynamic programming.

6.3.4 Lagrangian relaxation

The final topic for future research is the use of *Lagrangian relaxation* [Fis04, Lem07]. Let $x \in \mathbb{R}^n$. Consider the MILP

$$\begin{aligned} z = \quad \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & Dx \leq e \\ & x \geq 0 \text{ and integral,} \end{aligned}$$

where $b \in \mathbb{R}^m, e \in \mathbb{R}^k, A \in \mathbb{R}^{m \times n}$, and $D \in \mathbb{R}^{k \times n}$. The Lagrangian problem of the above MILP is

$$\begin{aligned} z_D(\lambda) = \quad \min \quad & c^T x + \lambda(Ax - b) \\ \text{s.t.} \quad & Dx \leq e \\ & x \geq 0 \text{ and integral,} \end{aligned}$$

where $\lambda = (\lambda_1, \ldots, \lambda_m)$ is a vector of Lagrange multipliers. In [Fis04], it is shown how to choose appropriately λ in order to approach z. From data of preliminary experiments, we believe that the MILP model could be solved much faster if some of the constraints are relaxed by moving them into the objective function. For example, we could readily relax the balance constraints (2.1b),(2.1c),(2.1d), and the stockpile constraints (2.1i),(2.1j), without affecting the binary variables. The relaxation of the feasible move constraints (2.1g) and (2.1h) would greatly reduce the problem size.

Bibliography

[Aig07] Martin Aigner. *A Course in Enumeration*, volume 238 of *Graduate Texts in Mathematics*. Springer-Verlag, 2007. → pages 29, 30

[Bel52] Richard E Bellman. The theory of dynamic programming. *Proceedings of the National Academy of Sciences of the United States of America*, 38(8):716–719, 1952. → pages 60

[BS05] Cyril Banderier and Sylviane Schwer. Why delannoy numbers? *Journal of Statistical Planning and Inference*, 135(1):40–54, 2005. → pages 33

[CGF89] E P Chew, C J Goh, and T F Fwa. Simultaneous optimization of horizontal and vertical alignments for highways. *Transportation Research Part B: Methodological*, 23(5):315–329, 1989. → pages 2, 4

[CL06] Juey-Fu Cheng and Yusin Lee. Model for three-dimensional highway alignment. *Journal of Transportation Engineering*, 132(12):913–920, 2006. → pages 4

[CLRS09] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 3 edition, Jan 2009. → pages 22, 34

[Dan98] George B Dantzig. *Linear Programming and Extensions*. Princeton Landmarks in Mathematics. Princeton University Press, 1998. → pages 36

[DM02] Elizabeth D Dolan and Jorge J Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002. → pages 48, 49

[Eas87] Said M Easa. Earthwork allocations with nonconstant unit costs. *Journal of Construction Engineering and Management*, 113(1):34–50, March 1987. → pages 7, 9, 59

[Eas88a] Said M Easa. Earthwork allocations with linear unit costs. *Journal of Construction Engineering and Management*, 114(4):641–655, December 1988. → pages 7, 9

[Eas88b] Said M Easa. Selection of roadway grades that minimize earthwork cost using linear programming. *Transportation Research. Part A: general*, 22(2):121–136, 1988. → pages 4

[Fis04] M Fisher. The Lagrangian relaxation method for solving integer programming problems. *Management science*, 50(12):1861–1871, December 2004. → pages 61

[Fou05] R Fourer. Linear programming - software survey. *Operations Research / Management Science Today*, (3):46, 2005. → pages 35, 36

[Fwa06] T F. Fwa. *The Handbook of Highway Engineering*. CRC Press, 1 edition, September 2006. → pages 43

[GH08] Nicholas J. Garber and Lester A. Hoel. *Traffic and Highway Engineering*. CL-Engineering, 4 edition, June 2008. → pages 4

[Gom58] Ralph E Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64:275–278, 1958. → pages 36

[JH90] A Jayawardane and F Harris. Further development of integer programming in earthwork optimization. *Journal of Construction Engineering and Management*, 116(1):18–34, Mar 1990. → pages 7

[JK06] Manoj Kumar Jha and Euncheol Kim. Highway route optimization based on accessibility, proximity, and land-use changes. *Journal of Transportation Engineering*, 132(5):435–439, May 2006. → pages 3

[JS03] J Jong and P Schonfeld. An evolutionary model for simultaneously optimizing three-dimensional highway alignments. *Transportation Research Part B: Methodological*, 37(2):107–128, February 2003. → pages 4

[JSJ06] Manoj Kumar Jha, Paul Schonfeld, and J-C Jong. *Intelligent Road Design*, volume 19. WIT Press, 2006. → pages 2, 3

[KL10] Valentin R Koch and Yves Lucet. A note on: Spline technique for modeling roadway profile to minimize earthwork cost. *Journal of Industrial and Management Optimization*, 6(2):393–400, May 2010. → pages 4, 59, 60

[Lay99] M G Lay. *Ways of the World: A History of the World's Roads and of the Vehicles that Used Them*. Rutgers University Press, 1999. → pages 1

[Lem07] Claude Lemaréchal. The omnipresence of Lagrange. *Annals of Operations Research*, 153(1):9–27, September 2007. → pages 61

[LTL09] Yusin Lee, You-Ren Tsou, and Hsiao-Liang Liu. Optimization method for highway horizontal alignment design. *Journal of Transportation Engineering*, 135(4):217–224, Jan 2009. → pages 3

[MA04] Ahmad A Moreb and M Aljohani. Quadratic representation for roadway profile that minimizes earthwork cost. *Journal of Systems Science and Systems Engineering*, 13(2):245–252, April 2004. → pages 59

[Mak99] K Makanae. An application of parametric curves to highway alignment. *Journal of Civil Engineering Information Processing System*, 8:231–238, Jan 1999. → pages 4

[McA24] John Loudon McAdam. *Remarks on the Present System of Road Making*. London, Longman, Hurst, Rees, Orme, Brown, and Green, 1824. → pages 1

[MKV02] S Mäki, R Kalliola, and K Vuorinen. Road construction in the Peruvian Amazon: process, causes and consequences. *Environmental Conservation*, 28(03):199–214, 2002. → pages 2

[Mor96] Ahmad A Moreb. Linear programming model for finding optimal roadway grades that minimize earthwork cost. *European Journal of Operational Research*, 93(1):148–154, August 1996. → pages 4, 59

[Mor09] Ahmad A Moreb. Spline technique for modeling roadway profile to minimize earthwork cost. *Journal of Inustrual and Management Optimization*, 5(2):275–283, 2009. → pages 4, 59, 60

[MS81] R Mayer and R Stark. Earthmoving logistics. *Journal of the Construction Division*, 107(2):297–312, Jun 1981. → pages ii, 6, 7, 11, 13, 57, 58

[PS98] Christos H Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Dover Publications, 1998. → pages 36

[SAV$^+$03] Robert Sulanke, Jose Arregui, Danijela Vujić, Nelson Castañeda, and David Callan. Another description of the central delannoy numbers: 10894. *The American Mathematical Monthly*, 110(5):443–444, May 2003. → pages 31

[Stu09] Les Stuart. Transportation in Canada 2008 - an overview. *Transport Canada*, pages 1–28, Jun 2009. → pages 1, 2

[Trb90] Trb. State of the art report 8: Guide to earthwork construction. *Transportation Research Board*, pages 1–119, Feb 1990. → pages 13

[Tri87] D Trietsch. A family of methods for preliminary highway alignment. *Transportation Science*, 21(1):17–25, February 1987. → pages 3

[Wei02] Eric W. Weisstein. *CRC Concise Encyclopedia of Mathematics.* CRC Press, December 2002. → pages 31, 33

Appendix A

Tables

Each design in Table A.1 is used with 2 access roads, and with 4 access roads for the same block configuration. An exception are the designs C and G with 20 blocks, where the use of 4 access roads was too hard to solve. Problems 23 and 24 are configurations were the blocks are mostly clustered together in adjacent sections.

The data for the problems is available upon request. Please contact the author at valentin.koch@gmail.com.

Table A.1: Problem configurations

Problem	Design	Length (km)	Sections	Access roads	Blocks
1,25	A	4	101	2,4	5
2,26	B	40	201	2,4	5
3,27	C	1	51	2,4	5
4,28	D	1	100	2,4	5
5,29	E	1	52	2,4	5
6,30	F	1	103	2,4	5
7,31	G	9	100	2,4	5
8,32	H	9	201	2,4	5
9,33	A	4	101	2,4	10
10,34	B	40	201	2,4	10
11,35	C	1	51	2,4	10
12,36	D	1	100	2,4	10
13,37	E	1	52	2,4	10
14,38	F	1	103	2,4	10
15,39	G	9	100	2,4	10
16,40	H	9	201	2,4	10
17,41	A	4	101	2,4	20
18	C	1	51	2	20
19,42	D	1	100	2,4	20
20,43	E	1	52	2,4	20
21,44	F	1	103	2,4	20
22	G	9	100	2	20
23	H	9	201	2	5
24	F	1	103	2	10

Table A.2: Statistics for objective function error with $d_{\max}$

ϵ		$d_{\max} = 5$	$d_{\max} = 25$	$d_{\max} = 100$	$d_{\max} = m_s$
0.01	Minimum	0.03344	-0.00907	-0.00351	0.00000
	Lower quartile	0.21869	0.00000	0.00000	0.00000
	Median	0.34841	0.00000	0.00000	0.00000
	Upper quartile	0.60876	0.04625	0.00000	0.00000
	Maximum	1.08125	0.18926	0.00000	0.00000
	Lower conf. for med.	0.23842	0.00000	0.00000	0.00000
	Upper conf. for med.	0.45841	0.01304	0.00000	0.00000
0.05	Minimum	-0.01434	-0.02768	-0.04701	0.00000
	Lower quartile	0.22466	0.00000	0.00000	0.00000
	Median	0.40394	0.00000	0.00000	0.00000
	Upper quartile	0.60948	0.03376	0.00000	0.00000
	Maximum	1.08494	0.20007	0.01690	0.00000
	Lower conf. for med.	0.30959	0.00000	0.00000	0.00000
	Upper conf. for med.	0.49830	0.00828	0.00000	0.00000

Table A.3: Statistics for solving time with $d_{\max}$

ϵ		$d_{\max} = 5$	$d_{\max} = 25$	$d_{\max} = 100$	$d_{\max} = m_s$
0.01	Minimum	0.00000	0.00016	0.00051	1.00000
	Lower quartile	0.03820	0.11630	0.59407	1.00000
	Median	0.10172	0.26389	0.99488	1.00000
	Upper quartile	0.16049	0.54474	1.00000	1.00000
	Maximum	0.44444	1.42857	1.50000	1.00000
	Lower conf. for med.	0.06544	0.13677	0.87444	1.00000
	Upper conf. for med.	0.13801	0.39101	1.00000	1.00000
0.05	Minimum	0.00000	0.00968	0.21221	1.00000
	Lower quartile	0.03800	0.09709	0.86429	1.00000
	Median	0.06395	0.13793	1.00000	1.00000
	Upper quartile	0.15395	0.28175	1.00000	1.00000
	Maximum	0.33205	1.27652	1.74622	1.00000
	Lower conf. for med.	0.03800	0.09709	0.96588	1.00000
	Upper conf. for med.	0.09310	0.18435	1.00000	1.00000

Table A.4: Histogram statistics for 1% tolerance

	direct	direct x°	warmstart	warmstart $\epsilon, 25$	warmstart $\epsilon, 5$
Solved	30	35	35	36	33
Min	-0.00674	0.00000	-0.00674	-0.00198	-0.00674
1^{st} qrt.	0.00041	0.00000	0.00000	0.00000	0.00000
Median	0.02386	0.01878	0.01878	0.02232	0.01696
3^{rd} qrt.	0.05308	0.04827	0.05051	0.05079	0.04658
Max	0.17902	0.17902	0.18987	0.17902	0.15804
Mean	0.04341	0.03950	0.04122	0.04088	0.03265
Std. dev	0.05429	0.05132	0.05519	0.05154	0.04227
Skew	1.13166	1.29910	1.35167	1.24205	1.34292

Table A.5: Histogram statistics for 5% tolerance

	direct	direct x°	warmstart	warmstart $\epsilon, 25$	warmstart $\epsilon, 5$
Solved	36	42	43	43	39
Min	-0.01722	0.00000	-0.02907	-0.02907	-0.04251
1^{st} qrt.	0.00000	0.00000	0.00000	0.00000	-0.00115
Median	0.01537	0.00000	0.00000	0.00000	0.00615
3^{rd} qrt.	0.04733	0.04055	0.04290	0.04290	0.03903
Max	0.17873	0.17815	0.18987	0.18987	0.15804
Mean	0.03865	0.03176	0.03309	0.03309	0.02331
Std. dev	0.05471	0.05305	0.05527	0.05526	0.04658
Skew	1.18703	1.46810	1.35854	1.35773	1.17794

Table A.6: CPU time in seconds for $\varepsilon = 0.01$, problems 1-22

Problem	direct	suc-	direct x°	suc-	warmstart	suc-	warmstart ϵ	suc-	$d^k_{\max}$	warmstart ϵ	suc-	$d^k_{\max}$
		cess		cess		cess	$(25, 10)$	cess		$(5, 5)$	cess	
1	79	1	77	1	82	1	73	1	250	68	1	125
2	911	1	1199	1	986	1	2622	1	250	3249	1	625
3	21	1	16	1	30	1	38	1	250	37	1	125
4	26	1	18	1	23	1	9	1	25	11	1	25
5	14	1	15	1	15	1	12	1	25	14	1	25
6	55	1	21	1	24	1	55	1	250	29	1	125
7	14	1	14	1	18	1	23	1	250	26	1	125
8	316	1	316	1	332	1	357	1	250	285	1	125
9	7104	1	1603	1	1507	1	1380	1	250	1171	1	125
10	43200	0	22338	1	43200	0	17274	1	250	18214	1	625
11	603	1	412	1	793	1	816	1	250	1472	1	125
12	2936	1	7522	1	1528	1	28593	1	250	3473	1	125
13	606	1	134	1	100	1	16	1	25	49	1	25
14	107860	0	43256	0	468	1	540	1	250	43231	0	125
15	376	1	146	1	206	1	285	1	250	223	1	125
16	15777	0	4864	1	5533	1	5629	1	250	11257	1	625
17	43200	0	84261	0	43200	0	87130	0	250	43337	0	5
18	43200	0	43200	0	43200	0	43200	0	25	43200	0	125
19	43324	0	43335	0	44453	0	43224	0	25	43176	0	5
20	43200	0	43200	0	43200	0	43200	0	25	43200	0	125
21	28	1	22	1	49	1	45	1	250	56	1	125
22	490	1	692	1	506	1	549	1	250	576	1	625

Table A.7: CPU time in seconds for $\varepsilon = 0.01$, problems 23-44

Problem	direct	suc-	direct x°	suc-	warmstart	suc-	warmstart ϵ	suc-	$d^k_{\max}$	warmstart ϵ	suc-	$d^k_{\max}$
		cess		cess		cess	$(25, 10)$	cess		$(5, 5)$	cess	
23	2	1	2	1	4	1	5	1	250	2	1	25
24	184	1	17	1	22	1	9	1	25	28	1	125
25	3	1	3	1	4	1	2	1	25	3	1	25
26	24	1	25	1	28	1	9	1	25	12	1	25
27	17	1	16	1	18	1	8	1	25	10	1	25
28	302	1	298	1	315	1	337	1	250	537	1	625
29	8531	1	15307	1	5369	1	10121	1	250	9506	1	125
30	43200	0	43200	0	43200	0	43200	0	25	43200	0	625
31	33	1	20	1	36	1	15	1	25	16	1	25
32	328	1	188	1	207	1	39	1	25	403	1	125
33	91	1	31	1	40	1	35	1	250	18	1	25
34	31175	1	22377	1	22458	1	8180	1	250	11565	1	125
35	197	1	182	1	196	1	38	1	25	54	1	25
36	43200	0	6486	1	7245	1	7362	1	250	3874	1	125
37	43200	0	43200	0	1536	1	7163	1	250	6891	1	125
38	2200	1	1425	1	2221	1	1430	1	250	43256	0	5
39	43200	0	9448	1	9628	1	11723	1	250	43342	0	25
40	43327	0	12576	1	43259	0	43202	0	250	43176	0	5
41	43200	0	44587	0	48538	0	49588	0	250	43200	0	125
42	56509	0	43208	0	43207	0	43200	0	25	43208	0	125
43	319	1	270	1	383	1	353	1	250	586	1	625
44	268	1	173	1	215	1	221	1	250	222	1	125

Table A.8: CPU time in seconds for $\varepsilon = 0.05$, problems 1-22

Problem	direct	suc- cess	direct x°	suc- cess	warmstart	suc- cess	warmstart ϵ $(25, 10)$	suc- cess	$d^k_{\max}$	warmstart ϵ $(5, 5)$	suc- cess	$d^k_{\max}$
1	59	1	19	1	23	1	26	1	250	40	1	125
2	886	1	601	1	1013	1	1044	1	250	107	1	25
3	20	1	14	1	29	1	30	1	250	38	1	125
4	27	1	18	1	23	1	9	1	25	15	1	25
5	14	1	15	1	13	1	13	1	25	15	1	25
6	56	1	19	1	23	1	8	1	25	12	1	25
7	14	1	14	1	18	1	8	1	25	14	1	25
8	323	1	322	1	338	1	367	1	250	386	1	125
9	1232	1	182	1	150	1	163	1	250	242	1	125
10	43200	0	2414	1	3356	1	1051	1	25	885	1	25
11	609	1	411	1	801	1	809	1	250	1559	1	125
12	235	1	145	1	155	1	30	1	25	53	1	25
13	596	1	134	1	101	1	88	1	25	61	1	25
14	1862	1	140	1	155	1	31	1	25	840	1	125
15	383	1	150	1	169	1	33	1	25	62	1	25
16	43200	0	4972	1	5635	1	5755	1	250	8530	1	250
17	43200	0	5967	1	1416	1	1738	1	250	43248	0	5
18	43200	0	43200	0	18899	1	19198	1	25	6564	1	25
19	33900	1	43338	0	43318	0	43308	0	250	42989	0	5
20	43200	0	1187	1	1295	1	636	1	25	3106	1	25
21	28	1	22	1	27	1	32	1	250	39	1	125
22	182	1	181	1	189	1	30	1	25	69	1	25

Table A.9: CPU time in seconds for $\varepsilon = 0.05$, problems 23-44

Problem	direct	suc- cess	direct x°	suc- cess	warmstart	suc- cess	warmstart ϵ $(25, 10)$	suc- cess	$d^k_{\max}$	warmstart ϵ $(5, 5)$	suc- cess	$d^k_{\max}$
23	2	1	2	1	3	1	2	1	25	3	1	25
24	154	1	18	1	21	1	8	1	25	12	1	25
25	3	1	3	1	4	1	2	1	25	3	1	25
26	25	1	26	1	28	1	8	1	25	14	1	25
27	17	1	16	1	19	1	8	1	25	12	1	25
28	309	1	305	1	323	1	343	1	250	686	1	625
29	186	1	169	1	204	1	228	1	250	351	1	125
30	5154	1	5117	1	5169	1	151	1	25	319	1	25
31	33	1	20	1	27	1	15	1	25	18	1	25
32	189	1	192	1	212	1	37	1	25	55	1	25
33	91	1	31	1	37	1	24	1	25	22	1	25
34	1536	1	198	1	214	1	41	1	25	60	1	25
35	200	1	187	1	200	1	37	1	25	64	1	25
36	14951	1	6628	1	7412	1	7513	1	250	5244	1	125
37	9991	1	1244	1	1418	1	1533	1	250	5606	1	125
38	1846	1	330	1	307	1	302	1	25	43051	0	5
39	43200	0	1130	1	1289	1	2144	1	250	43200	0	25
40	8412	1	1859	1	2333	1	3782	1	250	42900	0	5
41	43200	0	1732	1	5447	1	409	1	25	5617	1	25
42	57882	0	6651	1	7637	1	36483	1	25	38204	1	25
43	327	1	276	1	364	1	392	1	250	276	1	125
44	273	1	167	1	211	1	64	1	25	46	1	25

Table A.10: Objective value in $ for direct solve, $\varepsilon = 0.01$

Problem	Objective	success	ε	LP mass value
1	454426	1	0.00680	455181
2	9351412	1	0.00549	9974094
3	316359	1	0.00000	329040
4	489574	1	0.00446	505297
5	60684	1	0.00833	68898
6	209271	1	0.00540	211057
7	77114005	1	0.00198	76961263
8	56893108	1	0.00714	56926279
9	455163	1	0.00338	472891
10	10000000	0	0.08160	10204508
11	376427	1	0.00123	394819
12	493254	1	0.00982	519193
13	60687	1	0.00396	72079
14	212554	0	0.01895	212554
15	77114005	1	0.00198	76961263
16	0	0	1.00000	56975485
17	0	0	1.00000	477309
18	0	0	1.00000	516378
19	70262	0	0.02304	79597
20	0	0	1.00000	217812
21	451922	1	0.00983	448897
22	9351412	1	0.00966	9512758
23	297389	1	-0.00000	300402
24	482635	1	0.00000	505297
25	60253	1	0.00124	67322
26	209129	1	0.00559	209244
27	61940820	1	0.00000	61940820
28	55096639	1	0.00737	55096639
29	452486	1	0.00905	468367
30	9480000	0	0.04150	9620263
31	297389	1	0.00000	339047
32	484506	1	0.00539	482671
33	60686	1	0.00395	68391
34	210088	1	0.01000	211502
35	61940820	1	0.00000	61940820
36	56100000	0	0.02480	55132871
37	454000	0	0.01480	475342
38	300248	1	0.00952	360154
39	0	0	1.00000	488042
40	63540	0	0.04868	77395
41	0	0	1.00000	217519
42	0	0	1.00000	64716813
43	56487153	1	0.00000	56487153
44	208258	1	0.00073	214188

Table A.11: Objective value in $ for direct solve with x°, $\varepsilon = 0.01$

Problem	Objective	success	ε	LP mass value
1	454426	1	0.00861	455181
2	9351412	1	0.00366	9974094
3	316359	1	0.00205	329040
4	488986	1	0.00327	505297
5	60684	1	0.00813	68898
6	209271	1	0.00627	211057
7	76961263	1	0.00000	76961263
8	56926279	1	0.00771	56926279
9	455163	1	0.00584	472891
10	9498097	1	0.00866	10204508
11	376427	1	0.00591	394819
12	493254	1	0.00995	519193
13	60687	1	0.00396	72079
14	212554	0	0.01170	212554
15	76961263	1	0.00000	76961263
16	56975485	1	0.00857	56975485
17	462423	0	0.03118	477309
18	495000	0	0.01470	516378
19	70262	0	0.10922	79597
20	218000	0	0.04520	217812
21	448897	1	0.00316	448897
22	9351412	1	0.00000	9512758
23	297389	1	0.00000	300402
24	482635	1	0.00092	505297
25	60253	1	0.00124	67322
26	209244	1	0.00614	209244
27	61940820	1	0.00000	61940820
28	55096639	1	0.00737	55096639
29	452486	1	0.00933	468367
30	9430000	0	0.03620	9620263
31	297389	1	0.00000	339047
32	482671	1	0.00160	482671
33	60687	1	0.00396	68391
34	210088	1	0.00993	211502
35	61940820	1	0.00000	61940820
36	55132871	1	0.00802	55132871
37	455000	0	0.01560	475342
38	300248	1	0.00952	360154
39	485640	1	0.00771	488042
40	63540	1	0.00821	77395
41	217519	0	0.04395	217519
42	64716813	0	0.04289	64716813
43	56487153	1	0.00000	56487153
44	210165	1	0.00975	214188

Table A.12: Objective value in \$ for warmstart ($d^0_{\max} = 25, \alpha = 10$), $\varepsilon = 0.01$

Problem	Objective	success	ε	LP mass value
1	454426	1	0.00843	455181
2	9351412	1	0.00000	9974094
3	316359	1	0.00000	329040
4	489574	1	0.00446	505297
5	60684	1	0.00833	68898
6	209271	1	0.00627	211057
7	76961263	1	0.00000	76961263
8	56982170	1	0.00869	56926279
9	455163	1	0.00452	472891
10	9500000	0	0.03160	10204508
11	376427	1	0.00346	394819
12	493254	1	0.00982	519193
13	60687	1	0.00396	72079
14	212554	1	0.00996	212554
15	77114005	1	0.00198	76961263
16	56975485	1	0.00857	56975485
17	459000	0	0.02130	477309
18	496000	0	0.01730	516378
19	64484	0	0.05960	79597
20	218000	0	0.04510	217812
21	451922	1	0.00983	448897
22	9351412	1	0.00984	9512758
23	297389	1	0.00000	300402
24	483187	1	0.00206	505297
25	60253	1	0.00124	67322
26	209244	1	0.00614	209244
27	61940820	1	0.00000	61940820
28	55096639	1	0.00737	55096639
29	452486	1	0.00999	468367
30	9430000	0	0.03620	9620263
31	297389	1	0.00000	339047
32	482671	1	0.00160	482671
33	60687	1	0.00396	68391
34	210088	1	0.00993	211502
35	61940820	1	0.00000	61940820
36	55132871	1	0.00802	55132871
37	451067	1	0.00795	475342
38	300248	1	0.00952	360154
39	485640	1	0.00771	488042
40	63540	0	0.04868	77395
41	217519	0	0.04395	217519
42	64716813	0	0.04289	64716813
43	56487153	1	0.00000	56487153
44	210165	1	0.00975	214188

Table A.13: Objective value in $ for warmstart with ϵ ($d^0_{\max} = 25, \alpha = 10$), $\varepsilon = 0.01$

Problem	Objective	success	ε	ϵ	LP mass value
1	454426	1	0.00929	0.01434	455181
2	9351412	1	0.00591	0.02938	9974094
3	316359	1	-0.00000	0.06244	329040
4	488986	1	0.00327	0.00328	505297
5	60684	1	0.00027	0.00840	68898
6	209271	1	0.00542	0.00631	211057
7	76961263	1	0.00000	0.00000	76961263
8	56982170	1	0.00869	0.00876	56926279
9	455163	1	0.00338	0.01598	472891
10	9498097	1	0.00859	0.04553	10204508
11	376427	1	0.00074	0.26417	394819
12	493254	1	0.00989	0.01203	519193
13	60687	1	0.00396	0.00398	72079
14	212554	1	0.00996	0.02209	212554
15	77114005	1	0.00198	0.00198	76961263
16	56975485	1	0.00857	0.00857	56975485
17	458003	0	0.02183	0.02183	477309
18	496000	0	0.01730	0.01730	516378
19	70234	0	0.10560	0.33415	79597
20	218000	0	0.04510	0.04510	217812
21	448897	1	0.00316	0.00317	448897
22	9351412	1	0.00966	0.02938	9512758
23	297389	1	0.00000	0.00000	300402
24	482635	1	0.00092	0.00092	505297
25	60253	1	0.00124	0.00124	67322
26	209244	1	0.00614	0.00618	209244
27	61940820	1	0.00000	0.00000	61940820
28	55096639	1	0.00737	0.00742	55096639
29	452486	1	0.00926	0.01119	468367
30	9480000	0	0.04150	0.04150	9620263
31	297389	1	0.00000	0.00000	339047
32	482671	1	0.00160	0.00161	482671
33	60748	1	0.00496	0.00498	68391
34	210088	1	0.00996	0.01024	211502
35	61940820	1	0.00000	0.00000	61940820
36	55132871	1	0.00802	0.00809	55132871
37	451067	1	0.00795	0.00802	475342
38	300320	1	0.00976	0.00986	360154
39	484188	1	0.00473	0.00473	488042
40	63540	0	0.04868	0.05154	77395
41	217519	0	0.04395	0.04858	217519
42	0	0	1.00000	1.00000	64716813
43	56487153	1	0.00000	0.00000	56487153
44	208258	1	0.00068	0.00106	214188

Table A.14: Objective value in $ for warmstart with ϵ ($d^0_{\max} = 5, \alpha = 5$), $\varepsilon = 0.01$

Problem	Objective	success	ε	ϵ	LP mass value
1	454426	1	0.00843	0.01434	455181
2	9351412	1	0.00593	0.02938	9974094
3	316359	1	0.00000	0.06244	329040
4	488986	1	0.00327	0.00328	505297
5	60684	1	0.00833	0.00840	68898
6	209271	1	0.00627	0.00631	211057
7	76961263	1	0.00000	0.00000	76961263
8	56576215	1	0.00157	0.00158	56926279
9	455163	1	0.00600	0.01598	472891
10	9498097	1	0.00952	0.04553	10204508
11	376427	1	0.00154	0.26417	394819
12	493254	1	0.00865	0.01203	519193
13	60687	1	0.00396	0.00398	72079
14	212554	0	0.01436	0.05611	212554
15	77114005	1	0.00198	0.00198	76961263
16	56902685	1	0.00730	0.00730	56975485
17	656851	0	0.05385	0.52950	477309
18	496000	0	0.01710	0.01761	516378
19	82236	0	0.06211	0.55954	79597
20	218000	0	0.04550	0.04771	217812
21	451922	1	0.00983	0.00993	448897
22	9351412	1	0.00000	0.02938	9512758
23	297389	1	-0.00000	0.00000	300402
24	483187	1	0.00206	0.00207	505297
25	60253	1	0.00124	0.00124	67322
26	208491	1	0.00255	0.00256	209244
27	61940820	1	0.00000	0.00000	61940820
28	54767023	1	0.00139	0.00140	55096639
29	452486	1	0.00905	0.01119	468367
30	9430000	0	0.03620	0.03755	9620263
31	297389	1	0.00000	0.00000	339047
32	483954	1	0.00425	0.00427	482671
33	60735	1	0.00474	0.00477	68391
34	210088	1	0.00974	0.01024	211502
35	61940820	1	0.00000	0.00000	61940820
36	55231802	1	0.00980	0.00989	55132871
37	451067	1	0.00795	0.00802	475342
38	403569	0	0.06654	0.60979	360154
39	538301	0	0.10452	1.00000	488042
40	77847	0	0.07685	0.45749	77395
41	218000	0	0.04510	0.04801	217519
42	64716813	0	1.00000	0.04482	64716813
43	56487153	1	-0.00000	0.00000	56487153
44	210165	1	0.00975	0.01023	214188

Table A.15: Objective value in \$ for direct solve, $\varepsilon = 0.05$

Problem	Objective	success	ε	LP mass value
1	457354	1	0.01905	455181
2	9351412	1	0.00549	9974094
3	316359	1	0.04771	329040
4	489574	1	0.00446	505297
5	60684	1	0.00833	68898
6	209271	1	0.00540	211057
7	77114005	1	0.00198	76961263
8	56893108	1	0.00714	56926279
9	458942	1	0.02107	472891
10	10000000	0	0.08160	10204508
11	376427	1	0.02126	394819
12	494620	1	0.01462	519193
13	61213	1	0.01252	72079
14	212554	1	0.01895	212554
15	77114005	1	0.00198	76961263
16	0	0	1.00000	56975485
17	0	0	1.00000	477309
18	0	0	1.00000	516378
19	70262	1	0.02951	79597
20	0	0	1.00000	217812
21	451922	1	0.00983	448897
22	9351412	1	0.02854	9512758
23	297389	1	-0.00000	300402
24	503968	1	0.04316	505297
25	60253	1	0.00124	67322
26	209129	1	0.00559	209244
27	61940820	1	0.00000	61940820
28	55096639	1	0.00737	55096639
29	453208	1	0.01264	468367
30	9556950	1	0.04943	9620263
31	297389	1	0.00000	339047
32	487393	1	0.01128	482671
33	60686	1	0.00395	68391
34	210088	1	0.01013	211502
35	61940820	1	0.00000	61940820
36	56082250	1	0.02481	55132871
37	468035	1	0.04392	475342
38	301565	1	0.01385	360154
39	0	0	1.00000	488042
40	63562	1	0.04901	77395
41	0	0	1.00000	217519
42	0	0	1.00000	64716813
43	56487153	1	0.00000	56487153
44	208258	1	0.00073	214188

Table A.16: Objective value in $ for direct solve with x°, $\varepsilon = 0.05$

Problem	Objective	success	ε	LP mass value
1	455181	1	0.01577	455181
2	9351412	1	0.02854	9974094
3	316359	1	0.04304	329040
4	505297	1	0.03544	505297
5	60684	1	0.00813	68898
6	211057	1	0.01468	211057
7	76961263	1	0.00000	76961263
8	56926279	1	0.00771	56926279
9	466699	1	0.03908	472891
10	9498097	1	0.04354	10204508
11	376427	1	0.03718	394819
12	500476	1	0.02615	519193
13	60687	1	0.00396	72079
14	212554	1	0.02161	212554
15	76961263	1	0.00000	76961263
16	56975485	1	0.00857	56975485
17	466407	1	0.03946	477309
18	516000	0	0.05600	516378
19	70262	0	0.10922	79597
20	217812	1	0.04523	217812
21	448897	1	0.00316	448897
22	9512758	1	0.04502	9512758
23	300402	1	0.01003	300402
24	505297	1	0.04573	505297
25	60253	1	0.00124	67322
26	209244	1	0.00614	209244
27	61940820	1	0.00000	61940820
28	55096639	1	0.00737	55096639
29	468367	1	0.04460	468367
30	9556950	1	0.04943	9620263
31	297389	1	0.00000	339047
32	482671	1	0.00160	482671
33	60687	1	0.00396	68391
34	211502	1	0.01675	211502
35	61940820	1	0.00000	61940820
36	55132871	1	0.00802	55132871
37	463946	1	0.03549	475342
38	301010	1	0.01203	360154
39	488042	1	0.01259	488042
40	63607	1	0.04969	77395
41	217519	1	0.04395	217519
42	64716813	1	0.04289	64716813
43	56487153	1	0.00000	56487153
44	214188	1	0.02871	214188

Table A.17: Objective value in $ for warmstart ($d^0_{\max} = 25, \alpha = 10$), $\varepsilon = 0.05$

Problem	Objective	success	ε	LP mass value
1	454426	1	0.01413	455181
2	9351412	1	0.00000	9974094
3	316359	1	0.04485	329040
4	505297	1	0.03544	505297
5	60684	1	0.00833	68898
6	211057	1	0.01468	211057
7	78347224	1	0.01769	76961263
8	56982170	1	0.00869	56926279
9	461811	1	0.02990	472891
10	9498097	1	0.04354	10204508
11	376427	1	0.00346	394819
12	497561	1	0.02044	519193
13	61213	1	0.01252	72079
14	212554	1	0.02161	212554
15	78347224	1	0.01769	76961263
16	56975485	1	0.00857	56975485
17	461947	1	0.03018	477309
18	495973	1	0.01731	516378
19	64484	0	0.05960	79597
20	218355	1	0.04761	217812
21	461946	1	0.03132	448897
22	9512758	1	0.04502	9512758
23	300402	1	0.01003	300402
24	505297	1	0.04573	505297
25	60253	1	0.00124	67322
26	209244	1	0.00614	209244
27	61940820	1	0.00000	61940820
28	55096639	1	0.00737	55096639
29	456838	1	0.02048	468367
30	9425696	1	0.03620	9620263
31	301197	1	0.01264	339047
32	482671	1	0.00160	482671
33	61213	1	0.01252	68391
34	211502	1	0.01675	211502
35	61940820	1	0.00000	61940820
36	55132871	1	0.00802	55132871
37	455873	1	0.01841	475342
38	312180	1	0.04738	360154
39	488042	1	0.01259	488042
40	63540	1	0.04868	77395
41	218062	1	0.04633	217519
42	64716813	1	0.04289	64716813
43	57840191	1	0.02339	56487153
44	210680	1	0.01254	214188

Table A.18: Objective value in \$ for warmstart with ϵ ($d^0_{\max} = 25, \alpha = 10$), $\varepsilon = 0.05$

Problem	Objective	success	ε	ϵ	LP mass value
1	454426	1	0.01413	0.01434	455181
2	9351412	1	0.00000	0.02938	9974094
3	316359	1	0.04485	0.06244	329040
4	505297	1	0.03544	0.03674	505297
5	60684	1	0.00833	0.00840	68898
6	211057	1	0.01468	0.01489	211057
7	78347224	1	0.00000	0.01801	76961263
8	56982170	1	0.00869	0.00876	56926279
9	461811	1	0.02990	0.03082	472891
10	9498097	1	0.02020	0.04553	10204508
11	376427	1	0.00346	0.26417	394819
12	497561	1	0.02044	0.02087	519193
13	61213	1	0.01252	0.01267	72079
14	212554	1	0.02161	0.02209	212554
15	78347224	1	0.00000	0.01801	76961263
16	56975485	1	0.00857	0.00857	56975485
17	461947	1	0.03018	0.03112	477309
18	495973	1	0.01709	0.01761	516378
19	64484	0	0.05960	0.06679	79597
20	218355	1	0.04761	0.04999	217812
21	461946	1	0.03132	0.03233	448897
22	9512758	1	0.04502	0.04714	9512758
23	300402	1	0.01003	0.01013	300402
24	505297	1	0.04573	0.04792	505297
25	60253	1	0.00124	0.00124	67322
26	209244	1	0.00614	0.00618	209244
27	61940820	1	0.00000	0.00000	61940820
28	55096639	1	0.00737	0.00742	55096639
29	456838	1	0.02048	0.02091	468367
30	9425696	1	0.03620	0.03756	9620263
31	301197	1	0.01264	0.01281	339047
32	482671	1	0.00160	0.00161	482671
33	61213	1	0.01252	0.01267	68391
34	211502	1	0.01675	0.01703	211502
35	61940820	1	0.00000	0.00000	61940820
36	55132871	1	0.00802	0.00809	55132871
37	455873	1	0.01841	0.01876	475342
38	312180	1	0.04738	0.04974	360154
39	488042	1	0.01259	0.01259	488042
40	63562	1	0.04901	0.05154	77395
41	218062	1	0.04633	0.04858	217519
42	64716813	1	0.04289	0.04482	64716813
43	57840191	1	0.02339	0.02395	56487153
44	210680	1	0.01216	0.01270	214188

Table A.19: Objective value in \$ for warmstart with ϵ ($d^0_{\max} = 5, \alpha = 5$), $\varepsilon = 0.05$

Problem	Objective	success	ε	ϵ	LP mass value
1	454426	1	0.01413	0.01434	455181
2	9351412	1	0.02854	0.02938	9974094
3	316359	1	0.04655	0.06244	329040
4	488986	1	0.00327	0.03674	505297
5	60684	1	0.00833	0.00840	68898
6	211057	1	0.01468	0.01489	211057
7	78347224	1	0.00000	0.01801	76961263
8	56576215	1	0.00157	0.00876	56926279
9	461811	1	0.02990	0.03082	472891
10	9498097	1	0.04340	0.04553	10204508
11	376427	1	0.03643	0.26417	394819
12	501295	1	0.02774	0.02087	519193
13	60687	1	0.00396	0.01267	72079
14	219628	1	0.04992	0.02209	212554
15	78499966	1	0.00195	0.01801	76961263
16	56975485	1	0.00857	0.00857	56975485
17	656851	0	0.05385	0.03112	477309
18	495973	1	0.01709	0.01761	516378
19	82236	0	0.06379	0.06679	79597
20	217881	1	0.04553	0.04999	217812
21	461946	1	0.03132	0.03233	448897
22	9351412	1	0.02811	0.04714	9512758
23	297389	1	-0.00000	0.01013	300402
24	488396	1	0.01271	0.04792	505297
25	60253	1	0.00124	0.00124	67322
26	208491	1	0.00255	0.00618	209244
27	61940820	1	0.00000	0.00000	61940820
28	57438725	1	0.04784	0.00742	55096639
29	465460	1	0.03863	0.02091	468367
30	9425696	1	0.03620	0.03756	9620263
31	297389	1	0.00000	0.01281	339047
32	490197	1	0.01693	0.00161	482671
33	60735	1	0.00474	0.01267	68391
34	211607	1	0.01724	0.01703	211502
35	61940820	1	0.00000	0.00000	61940820
36	55231802	1	0.00980	0.00809	55132871
37	455042	1	0.01662	0.01876	475342
38	403569	0	0.06654	0.04974	360154
39	560000	0	1.00000	0.01259	488042
40	77847	0	0.07685	0.05154	77395
41	217944	1	0.04581	0.04858	217519
42	64716813	1	0.04289	0.04482	64716813
43	57205059	1	0.01255	0.02395	56487153
44	218432	1	0.04759	0.01270	214188

Appendix B

Figures

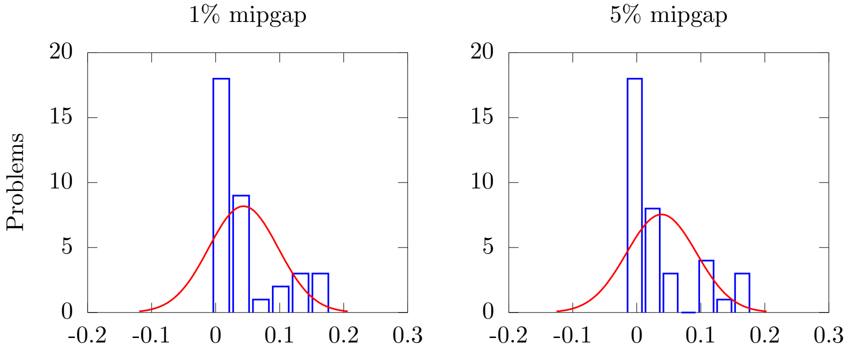


Figure B.1: Savings with direct solve, compared to mass block removal schedule

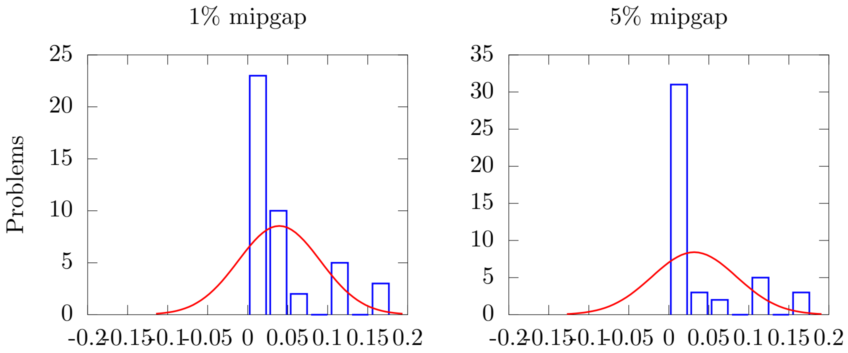


Figure B.2: Savings with direct solve and x°, compared to mass block removal schedule

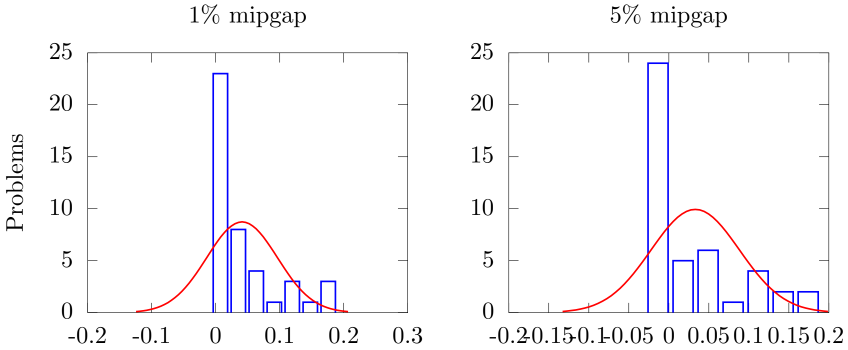


Figure B.3: Savings with warmstart ($d^0_{\max} = 25, \alpha = 10$), compared to mass block removal schedule

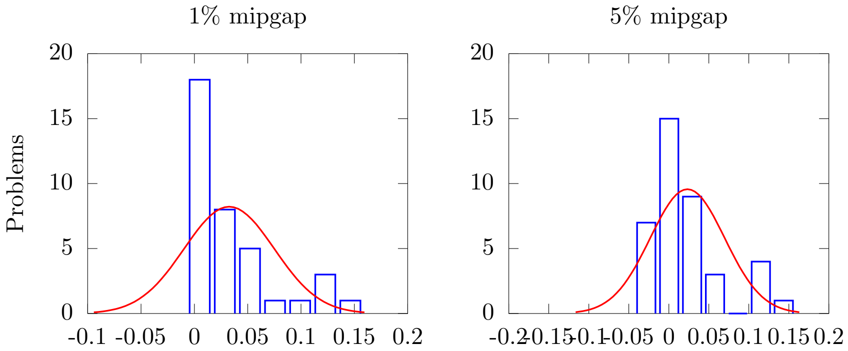


Figure B.4: Savings with warmstart and ϵ ($d^0_{\max} = 5, \alpha = 5$), compared to mass block removal schedule

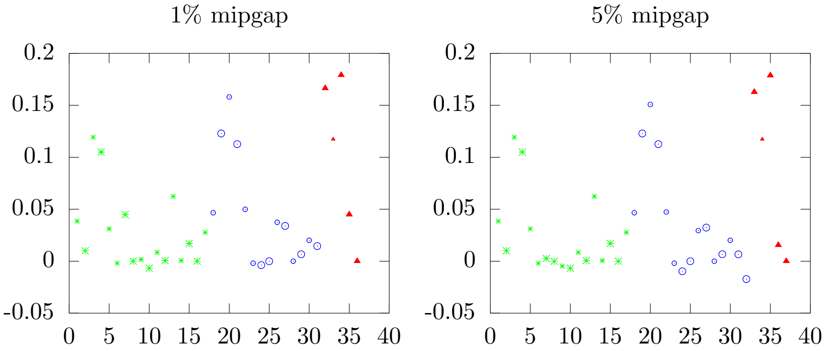


Figure B.5: Stratification of gains for direct solve; * are 5, ⊙ are 10, ▲ are 20 blocks; small symbols are 2, big symbols are 4 access roads; within the block groups, points are sorted from left to right in ascending order of section numbers

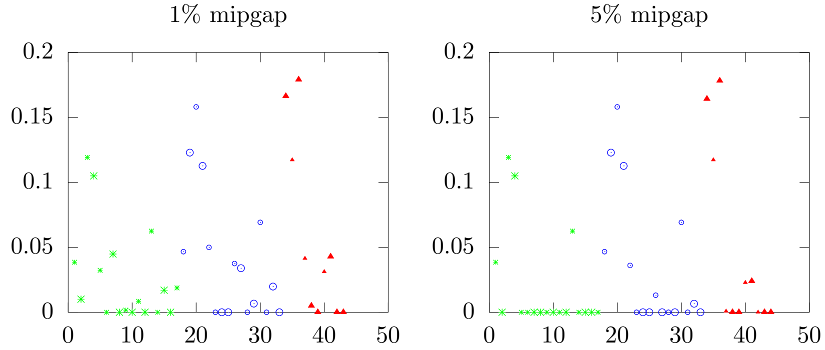


Figure B.6: Stratification of gains for direct solve with x°; green=5, blue=10, red=20 blocks; * are 5, ⊙ are 10, ▲ are 20 blocks; small symbols are 2, big symbols are 4 access roads; within the block groups, points are sorted from left to right in ascending order of section numbers

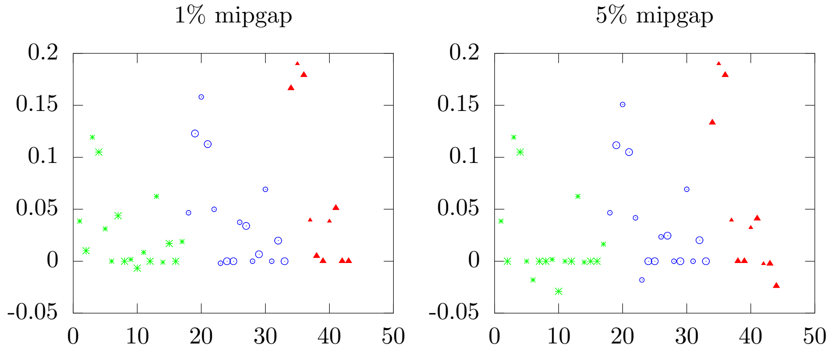


Figure B.7: Stratification of gains for warmstart ($d^0_{\max} = 25, \alpha = 10$); * are 5, ⊙ are 10, ▲ are 20 blocks; small symbols are 2, big symbols are 4 access roads; within the block groups, points are sorted from left to right in ascending order of section numbers

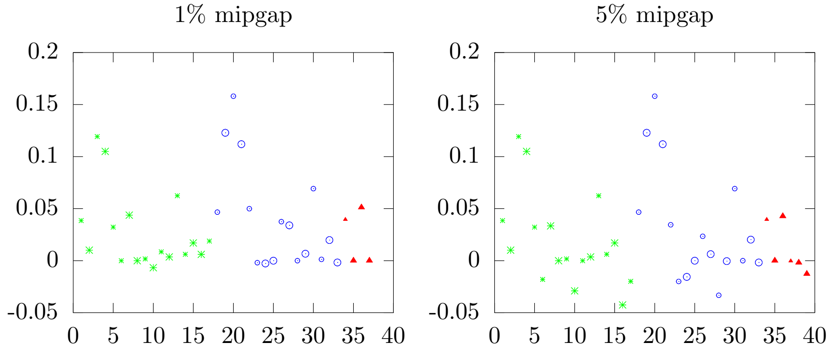


Figure B.8: Stratification of gains for warmstart with ϵ ($d^0_{\max} = 5, \alpha = 5$); * are 5, ⊙ are 10, ▲ are 20 blocks; small symbols are 2, big symbols are 4 access roads; within the block groups, points are sorted from left to right in ascending order of section numbers