PIMS

MITACS

Proceedings of the Seventh

PIMS-MITACS Graduate Industrial Math Modelling Camp

May 10–14, 2004
University of Victoria

Editors: Reinhard Illner (University of Victoria)
and David Leeming (University of Victoria)

FOREWORD BY THE PIMS DIRECTOR

The annual PIMS **Graduate Industrial Math Modelling Camp (GIMMC)** is held in one of the PIMS universities as part of the PIMS industrial programme. It is part of PIMS' commitment to providing training for young mathematical scientists who are either pursing careers in academia or in industry.

The goal of the GIMMC is to provide experience in the use of mathematical modelling as a problem-solving tool for graduate students in mathematics, applied mathematics, statistics and computer science. In addition to this it helps prepare them for the **Industrial Problem Solving Workshop** which takes place the week after GIMMC.

At the workshop students work together in teams, under the supervision of invited mentors. Each mentor poses a problem arising from an industrial or engineering application and guides his or her team of graduate students through a modelling phase to a resolution.

The Seventh GIMMC was held May 10–14, 2004, at the University of Victoria in British Columbia. This year it was co-sponsored by MITACS (The Mathematics of Information Technology and Complex Systems, a Network of Centres of Excellence).

There were six problems posed, with a total of 40 students in attendance. The students mainly came from across North America with 7 from the United States and 3 international students.

My sincere appreciation and gratitude goes to all the people involved in this workshop. In particular I would like to thank the organisers (Rex Westbrook, Kes Salkauskas), the local organising committee (David Leeming, Reinhard Illner, Pauline van den Driessche, Julie Zhou) and the industrial mentors (C. Sean Bohun, Tim Myers, Tobias Schaefer, Petra Berenbrink, Randall Pyke, Peter Ehlers).

I look forward to the 2005 GIMMC which will be held at the University of Lethbridge.

Dr. Ivar Ekeland, Director
Pacific Institute for the Mathematical Sciences

EDITOR'S PREFACE

From May 10 through May 14, 2004, forty-five graduate students and six mentors gathered at the University of Victoria for the Seventh Graduate Industrial Mathematical Modelling Camp (GIMMC). The students came from universities across Canada, the United States, Austria and Korea. The single Austrian and two Korean students were sponsored by their home countries.

It was rewarding to be part of the second GIMMC held at the University of Victoria, where the first one took place in 1998. Through GIMMC and the Industrial Problem Solving Workshop, held the following week at the University of British Columbia, the group had excellent exposure to a variety of industrial mathematics modelling problems.

The participants arrived on Sunday, May 9 and that evening were hosted at a welcoming reception at the UVic Graduate Centre. They were also given a brief tour of the UVic campus. Monday morning, the participants were officially welcomed to UVic by Dr. Martin Taylor, Vice President Research. This was followed by brief presentations of the problems by the six mentors. Immediately afterwards they broke into teams to tackle the problems.

Again this year an excellent group of mentors challenged and guided the students through the week. From Monday morning through Thursday evening the individual workshop groups and their mentors were hard at work. Computers and software needs were provided by one of the computing facilities on campus. On Friday morning the students presented their findings.

Formal write-ups of their progress were submitted some time after the workshop. These reports, after a refereeing process and some editorial work, form the rest of this document. They document remarkable progress considering that each project involved only a handful of graduate students with varying mathematical backgrounds and a total of three and a half days of effort.

A workshop of this size can only be successful with the combined effort of many. First, we would like to thank the mentors, without whom there could be no GIMMC. They were:

- Sean Bohun (Pennsylvania State University)
- Tim Myers (University of Cape Town and UBC)
- Tobias Schaefer (University of North Carolina at Chapel Hill)
- Petra Berenbrink (Simon Fraser University)
- Randall Pyke (University College of the Fraser Valley)
- Peter Ehlers (University of Calgary)

All of the mentors were excellent both leading up to and during the week; we cannot thank them enough for their efforts. Regrettably, no writeup from the team led by Peter Ehlers was available at publication time, and therefore their project is not documented in this report.

We would also like to express our appreciation to the staff at UVic who helped with the organization and behind the scenes work. The PIMS Administrative Assistant, Dil Bains, and the Computer Systems Administrator, Kelly Choo, were invaluable. We also acknowledge the UVic computing staff who provided helpful advice throughout the week. We are also grateful for the cheerful and expert support received from the PIMS staff, particularly Andrea Hook and Heather Jenkins. Thanks also to the other members of the UVic organizing committee—Pauline van den Driessche and Julie Zhou.

Finally, we thank the organizers Rex Westbrook and Ken Salkauskas from the University of Calgary for their many months of preparation leading to a very successful modelling camp.

Reinhard Illner and David Leeming, Co-Editors
Department of Mathematics and Statistics
University of Victoria

February 2005

Contents

Chapter 1

Optimal Design of Gas Burst Gene Gun

Participants: C. Sean Bohun (Mentor), Hooman Javidnia, Olga Krakovska.

PROBLEM STATEMENT: A gene gun consists of a tube that is configured to propel DNA-coated micro-particles, currently tungsten or gold, into a wide range of biological samples. There are a number of propulsion methods that have been investigated in the past including but not limited to electric discharge, gun powder explosions, bursts of helium gas and gas flow. One particular helium burst system (Bio Rad PDS 1000/He) propels a kapton disk coated with the micro-particles into a stopping screen. The momentum of the disk is transferred to the micro-particles which continue into the cellular structure of the target.

Our goal is to model the dynamics of the micro-particles as a first step in designing a hand held device using a CO_2 cartridge. The first challenge is to model the flow of the gas-particle suspension. There are natural limitations. For example, if the pressure is too high, the biological sample can be damaged. If the pressure is too low, the micro-particles will not penetrate the target. Is there an optimal pressure that maximizes the acceleration of the micro-particles? Is there any advantage in shaping the pressure pulse?

In the described system, the kapton disk is not mechanically guided into the stopping screen. As a result, the actual centre of impact of the micro-particles can change from one run to the next. The second challenge is to consider the addition of a nozzle which acts to simultaneously focus the flow as well as allowing one to change the velocity distribution of the exiting micro-particles.

1.1 Introduction

A fundamental problem faced by geneticists is the ability to modify a given segment of DNA and to study the effects of the induced genetic modification. There are several methods of introducing DNA into animal and plant cells. A gene gun is a device which can be used to accelerate DNA-coated micro-particles directly into living tissue [8, 9, 12]. The introduced particles are incorporated into the target cells and can bring about a change in the cell's genetic properties. In this work, we model the behaviour of such a device and search for ways in which to optimize the design.

The gas burst gene gun consists of a source of high pressure gas, a shock tube, and a nozzle. Rather than using a kapton disk to carry the micro-particles, they are instead placed on the interior wall of the shock tube near the entrance of the nozzle as shown in Figure 1.1. A short pressure pulse imparts a drag force on the micro-particles which accelerates them through the nozzle and subsequently into the target. Our specific modelling effort will suppose that the accelerating gas is helium so that an ideal gas law can be used which significantly simplifies the analysis. As for the micro-particles, we require a high density material that does not react with our biological sample. Many materials are available as a delivery particle including tungsten, gold, polystyrene, borosilicate and stainless steel [8]. Our model will assume the use of gold since this has the highest density and is available as essentially spherical particles.

Without the kapton disk, the underlying theory of operation of a gas burst gene gun is closely related to the concept of a cold spray [1, 3, 4, 5]. This is a process whereby an ultra fine metal powder is accelerated to supersonic velocities and deposited on the surface of the target material.

1.2 Modelling the Shock Tube

The underlying assumptions for the shock tube include:

- the diameter of the tube is much smaller than its length,
- viscosity of the the gas is ignored and no heat energy is exchanged with the walls of the tube,
- the tube is considered to be frictionless with respect to the flow of the gas.

The first assumption allows one to reduce the problem to one spatial dimension whereas the second and third assumptions allow us to assume that the gas flow is *isentropic* (adiabatic and frictionless). Under these assumptions the conservation of mass, momentum and energy can be expressed as:

$$\rho_t + (\rho v)_x = 0 \tag{1.1}$$

$$(\rho v)_t + \left(\rho v^2 + P\right)_x = 0 \tag{1.2}$$

$$E_t + (v\,(E+P))_x = 0 \tag{1.3}$$

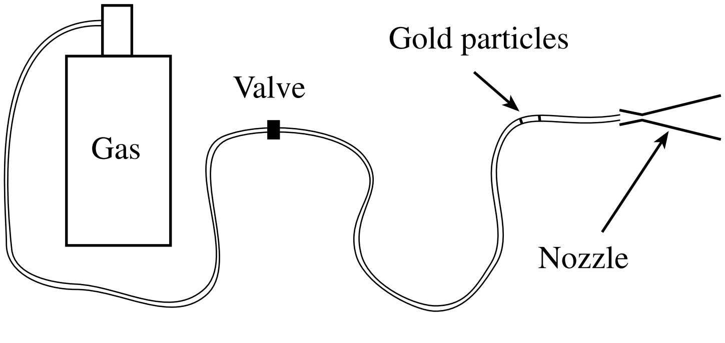

Figure 1.1: Gene gun geometry.

where ρ, v, and P respectively denote the density, speed and pressure of the gas. The variable E denotes the total energy of the gas per unit volume and can be written as the sum of gas' internal energy and its kinetic energy. In the case of an isentropic gas,

$$E = \frac{1}{2}\rho v^2 + \frac{1}{\gamma - 1}P \tag{1.4}$$

where γ is the ratio of a specific heats. For helium $\gamma = c_{\mathrm{p}}/c_{\mathrm{v}} = 5/3$.

To investigate the qualitative behaviour of the gas under experimental conditions the shock tube equations are nondimensionlized by making the following change of variables:

$$\begin{aligned} &\rho \to \mu\hat{\rho} \quad && P \to \lambda\hat{P} \quad && E \to \lambda\hat{E} \\ &x \to \xi\hat{x} \quad && t \to \tau\hat{t} \quad && v \to \frac{\xi}{\tau}\hat{v}. \end{aligned}$$

The reference scales μ, λ, ξ, and τ are the characteristic values of the respective density, pressure, length and duration of the shock tube problem under consideration. Under this change of variables we have equations (1.1-1.4) in nondimensionlized form (dropping hats):

$$\rho_t + (\rho v)_x = 0 \tag{1.5}$$

$$(\rho v)_t + \left(\rho v^2 + \frac{\lambda\tau^2}{\mu\xi^2}P\right)_x = 0 \tag{1.6}$$

$$E_t + [v(E+P)]_x = 0 \tag{1.7}$$

$$E = \frac{1}{2}\frac{\mu\xi^2}{\lambda\tau^2}\rho v^2 + \frac{1}{\gamma - 1}P\,. \tag{1.8}$$

From expression (1.4) we see that the nondimensional quantity $\lambda\tau^2/\mu\xi^2$, which characterizes the flow in the shock tube, is simply a measure of the ratio of the internal energy to the kinetic energy of the gas in the tube. Returning to experimental situation, the pulse pressure of the gene gun occurs over a typical interval of $\tau \approx 50\,\mathrm{ms}$ at a pressure of about four atmospheres ($\lambda \approx 4 \times 10^5\,\mathrm{Pa}$). If we assume that the tube contains He gas with a density of $\mu \approx 0.5\,\mathrm{kg/m^3}$, and that the tube is about $\xi = 2\,\mathrm{m}$ long then

$$\frac{\lambda\tau^2}{\mu\xi^2} \approx 500.$$

Since $\mu\xi^2/\lambda\tau^2 \ll 1$, most of the gas energy is internal rather than kinetic in nature and to first order expression (1.8) becomes $E \approx P/(\gamma - 1)$. Using the same argument we can simplify equation (1.6) to $P_x = 0$. This indicates that during its operation, the pressure should not vary spatially in the shock tube. By the isentropic assumption, the density is also constant along the tube. The above analysis indicates that over the duration of the pulse the conditions of the gas entering the nozzle (speed, pressure, density) will remain constant. In addition, the characteristics of the gas passing through the nozzle must be established by the time ($\tau \approx \xi\sqrt{\mu/\lambda} \lesssim 2\,\mathrm{ms}$).

Initial conditions for the numerical simulation of the shock tube include the following:

Temperature: The tube is in thermal equilibrium with room temperature, $T_0 = 300\,\mathrm{K}$.

Velocity: The gas in not initially moving, $v_0 = 0$.

Pressure: The pressure in the tube is initially given by

$$P(x,0) = \begin{cases} P_{\mathrm{L}} = 4\,\mathrm{atm}, & x < x_0 \\ P_{\mathrm{R}} = 1\,\mathrm{atm}, & x \geq x_0 \end{cases}$$

where x_0 is the location of the valve in the tube. For the simulations, the helium tank is located at $x = -4\,\mathrm{m}$, the valve is at $x_0 = 0\,\mathrm{m}$ and the nozzle is at $x = 1\,\mathrm{m}$.

Density: Density is determined from the pressure $P(x,0)$ with the ideal gas equation

$$P(x,0) = \rho(x,0)RT_0$$

where the ideal gas constant for He is $R = 8.314 \times 10^3\,\mathrm{J\,K^{-1}\,mol^{-1}}/4.003\,\mathrm{kg\,mol^{-1}}$.

1.3 Simulation of Shock Tube

To solve the shock tube equations we write the system (1.5–1.8) in the form of

$$\boldsymbol{u}_t + A\boldsymbol{u}_x = \boldsymbol{0} \tag{1.9}$$

where

$$\boldsymbol{u} = \begin{bmatrix} \rho \\ \rho v \\ E \end{bmatrix}, \qquad A = \begin{bmatrix} 0 & 1 & 0 \\ \dfrac{\gamma-3}{2}v^2 & -(\gamma-3)v & \gamma-1 \\ \dfrac{\gamma-1}{2}v^3 - (E+P)\dfrac{v}{\rho} & -(\gamma-1)v^2 - (E+P)\dfrac{1}{\rho} & \gamma v \end{bmatrix}. \tag{1.10}$$

At the edges of the spatial domain we assign a zero Neumann boundary condition which is consistent provided no disturbance in any of the variables has time to reach the boundary of the domain. The problem as described is known as Sod's problem [10] and an exact solution is well known and can be found in any good gas dynamics text for example [2].

In the last decade there has been rapid progress in the development of high resolution techniques for conservation laws. See for example the texts by Leveque [6, 7]. The technique chosen to solve Sod's problem was greatly influenced by the time constraints of the modelling camp as well as the background of the participants. Since the behaviour of the exact solution is known for this problem a numerical method was chosen that could reproduce this known solution with the least amount of coding. Because of these considerations, to solve the equation (1.9-1.10) numerically, we used the linearized Lax-Wendroff technique with the addition of some explicit second order dissipation. For convenience, we take a uniform grid with grid spacing $\Delta x = 0.01$ on the spatial axis and $\Delta t = 0.001$ on the temporal axis. The notation u_k^n describes the value of the variable u at the grid location $(k\Delta x, n\Delta t)$. With this notation the discretization scheme takes the form [13, 14]

$$\boldsymbol{u}_k^{n+1} = \boldsymbol{u}_k^n - \frac{R}{2}A^n\delta_0\boldsymbol{u}_k^n + \frac{R^2}{2}\left(A^n\right)^2\delta^2\boldsymbol{u}_k^n + \nu\frac{\Delta t}{(\Delta x)^2}\delta^2\boldsymbol{u}_k^n$$

where $R = \Delta t/\Delta x$, $A^n = A(\boldsymbol{u}^n)$, δ_0 is the centred difference operator

$$\delta_0\boldsymbol{u}_k^n = \boldsymbol{u}_{k+1}^n - \boldsymbol{u}_{k-1}^n\,,$$

and δ^2 is the second order difference operator

$$\delta^2\boldsymbol{u}_k^n = \boldsymbol{u}_{k+1}^n - 2\boldsymbol{u}_k^n + \boldsymbol{u}_{k-1}^n\,.$$

Since the leading term in the error associated with a smooth solution of the Lax-Wendroff scheme is dispersive [13], ν corrects for this by adding a controlled amount of dissipation. For the simulation we set $\nu = 0.001$. Of course the exact solution is *not* smooth and ν is used to eliminate the inevitable oscillations that are introduced whenever a second order scheme is used. By adding the viscosity "manually" there is the risk that small scale waves due to dispersion may be eliminated causing the solution to blow up. If the exact solution was not known then (1.9-1.10) would have been solved with a high resolution technique.

Figure 1.2 illustrates main characteristics of the solution. There is a shock in all three variables propagating at a speed of $v + c$, a contact discontinuity in the density moving at a speed of v and a rarefaction fan in all three variables moving at a speed of $v - c$. At the captured time index, the pressure shock has just reached the nozzle indicated by the dotted line at $x = 1\,\mathrm{m}$. As is apparent from the figure, this occurs at $t = 0.81\,\mathrm{ms}$. By this time, the contact discontinuity in the density has propagated approximately $34\,\mathrm{cm}$ from the valve at a speed of $v = 420\,\mathrm{m\,s^{-1}}$. At about $t = 2.3\,\mathrm{ms}$ the jump in the density reaches the nozzle entrance and the properties of the gas in the nozzle stabilize. These stabilized values are

$$\begin{aligned} P(x_0,t) &= 1.9\,\mathrm{atm} \\ \rho(x_0,t) &= 0.42\,\mathrm{kg\,m^{-3}} \\ v(x_0,t) &= 420\,\mathrm{m\,s^{-1}} \end{aligned}$$

and the temperature has dropped from $T_0 = 300\,\mathrm{K}$ to $T(x_0,t) = 221\,\mathrm{K}$. Consequently the speed of sound has also dropped to $c_{\text{in}} = 874\,\mathrm{m\,s^{-1}}$ and the Mach number of the gas at the entrance of the nozzle is $M_{\text{in}} = v(x_0,t)/c_{\text{in}} = 0.48$. These values will be used as the initial conditions of the nozzle.

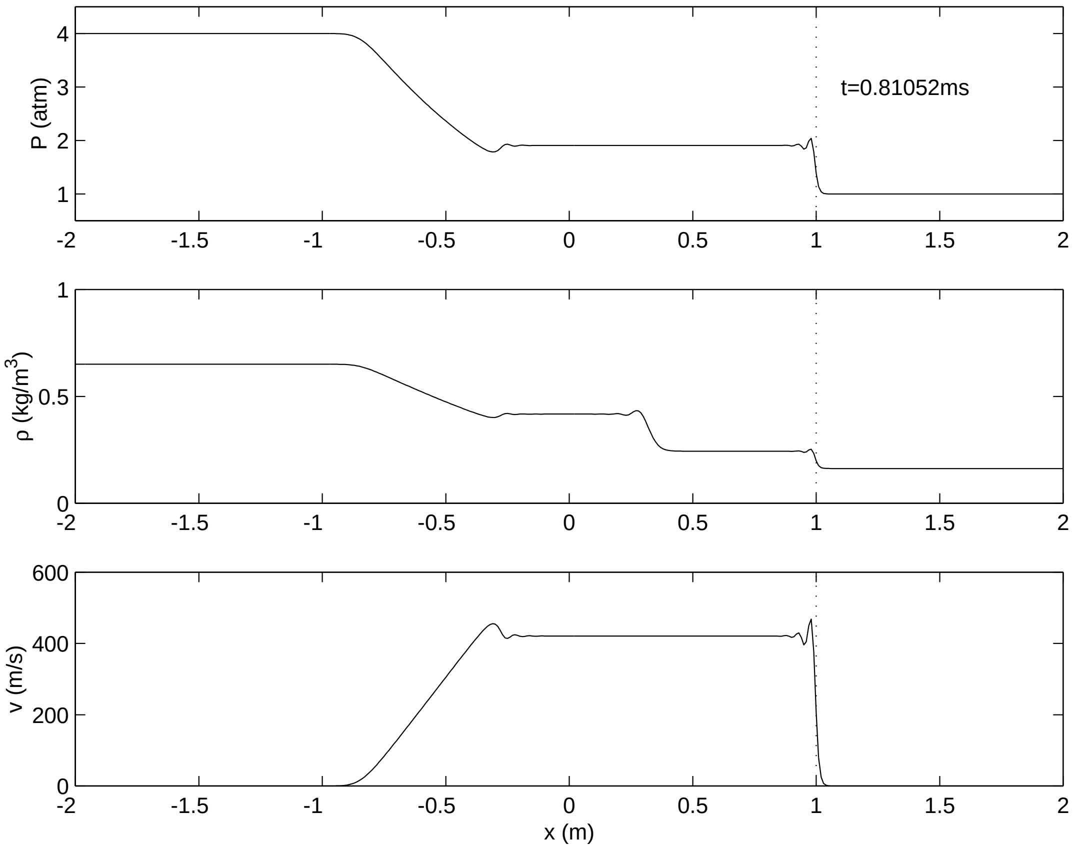

Figure 1.2: Shock tube variables when the pressure pulse reaches the nozzle.

1.4 Modelling the Nozzle

Similar to the shock tube, we begin by assuming that the gas flow through the nozzle is one-dimensional and isentropic. Figure 1.3 illustrates a typical nozzle with a radius that is piecewise linear. It will be apparent from the subsequent analysis that no such restriction will be required for a given nozzle design. We begin by illustrating how a nozzle is used to accelerate the gas particles.

An ideal, isentropic gas satisfies the additional relationship $P\rho^{-\gamma} = \Gamma_0 = \text{const}.$ As a result, the speed of sound of the gas is given by

$$c^2 \equiv \frac{\mathrm{d}P}{\mathrm{d}\rho} = \gamma\Gamma_0\rho^{\gamma-1} = \gamma\frac{P}{\rho} = \gamma RT. \tag{1.11}$$

For steady one-dimensional flow through the nozzle the continuity equation reduces to $\rho v A = \text{const}.$ where A is the cross sectional area at any point in the nozzle. Expressed in terms of differentials this is

$$\frac{\mathrm{d}\rho}{\rho} + \frac{\mathrm{d}v}{v} + \frac{\mathrm{d}A}{A} = 0. \tag{1.12}$$

In the same steady one-dimensional limit the conservation of momentum reduces to [11]

$$v\,\mathrm{d}v + \frac{1}{\rho}\,\mathrm{d}P = v\,\mathrm{d}v + \frac{c^2}{\rho}\,\mathrm{d}\rho = 0. \tag{1.13}$$

Eliminating $\mathrm{d}\rho/\rho$ from the relations (1.12) and (1.13) and rearranging gives the differential equation

$$\frac{\mathrm{d}A}{\mathrm{d}v} = \frac{A}{v}(M^2-1)\,, \tag{1.14}$$

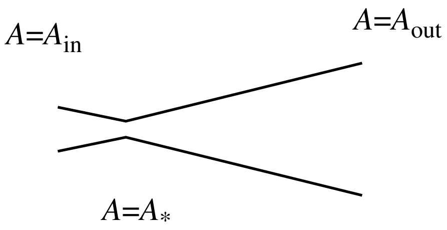

Figure 1.3: Nozzle geometry.

where the Mach number $M = v/c$. Expression (1.14) embodies the purpose of the nozzle. Incident gas particles approaching the nozzle at subsonic speeds $(v < c)$ are accelerated with the converging portion. If the particles reach the sonic speed $(v = c)$ at the throat of the nozzle then they can be further accelerated to supersonic speeds $(v > c)$ with the diverging portion. To summarize, the maximum possible exit velocity of the gas particles is attained by having subsonic flow in the converging part of the nozzle and supersonic flow in diverging part. At the throat $v = c$ or $M_* = 1$.

The dependence of pressure, temperature, and density can be written as a function of the Mach number. This is exhibited by combining the momentum expression (1.13) with the isentropic relationship $P\rho^{-\gamma} = \Gamma_0$. Differentiating,

$$\mathrm{d}P\rho^{-\gamma} - \gamma P\rho^{-\gamma-1}\,\mathrm{d}\rho = 0$$

and by using equation (1.13),

$$\mathrm{d}P\rho^{-1} = \gamma P\rho^{-2}\,\mathrm{d}\rho = -v\,\mathrm{d}v$$

or

$$v\,\mathrm{d}v + \gamma\Gamma_0\rho^{\gamma-2}\,\mathrm{d}\rho = 0.$$

Integrating and using both $P\rho^{-\gamma} = \Gamma_0$ and $P = \rho RT$ yields

$$\frac{v^2}{2} + \frac{\gamma}{\gamma-1}RT = \text{const.} \tag{1.15}$$

which is essentially a statement for conservation of energy within the nozzle.

Consider two regions within the nozzle. The entrance, characterized by values $(P_{\text{in}}, \rho_{\text{in}}, T_{\text{in}}, v_{\text{in}})$ and another position characterized by (P, ρ, T, v). Using the conservation of energy (1.15) and (1.11) gives the expression

$$\frac{T_{\text{in}}}{T} = \frac{1 + \dfrac{\gamma-1}{2}M^2}{1 + \dfrac{\gamma-1}{2}M_{\text{in}}^2}. \tag{1.16}$$

By using the ideal gas and isentropic flow expressions the density and pressure evolution is given by

$$\frac{\rho_{\text{in}}}{\rho} = \left(\frac{T_{\text{in}}}{T}\right)^{\frac{1}{\gamma-1}} \tag{1.17}$$

$$\frac{P_{\text{in}}}{P} = \left(\frac{T_{\text{in}}}{T}\right)^{\frac{\gamma}{\gamma-1}}. \tag{1.18}$$

The Mach number variation within the nozzle is determined by the nozzle geometry. This is established by using the conservation of mass $\rho v A = \text{const}$. Let M_* denote the Mach number of the gas at the throat of the nozzle where $A = A_*$. Then

$$\frac{A}{A_*} = \frac{\rho_* v_*}{\rho v} = \frac{\rho_*}{\rho_{\text{in}}}\frac{\rho_{\text{in}}}{\rho}\frac{v_*}{c_*}\left(\frac{T_*}{T_{\text{in}}}\frac{T_{\text{in}}}{T}\right)^{1/2}\frac{c}{v}.$$

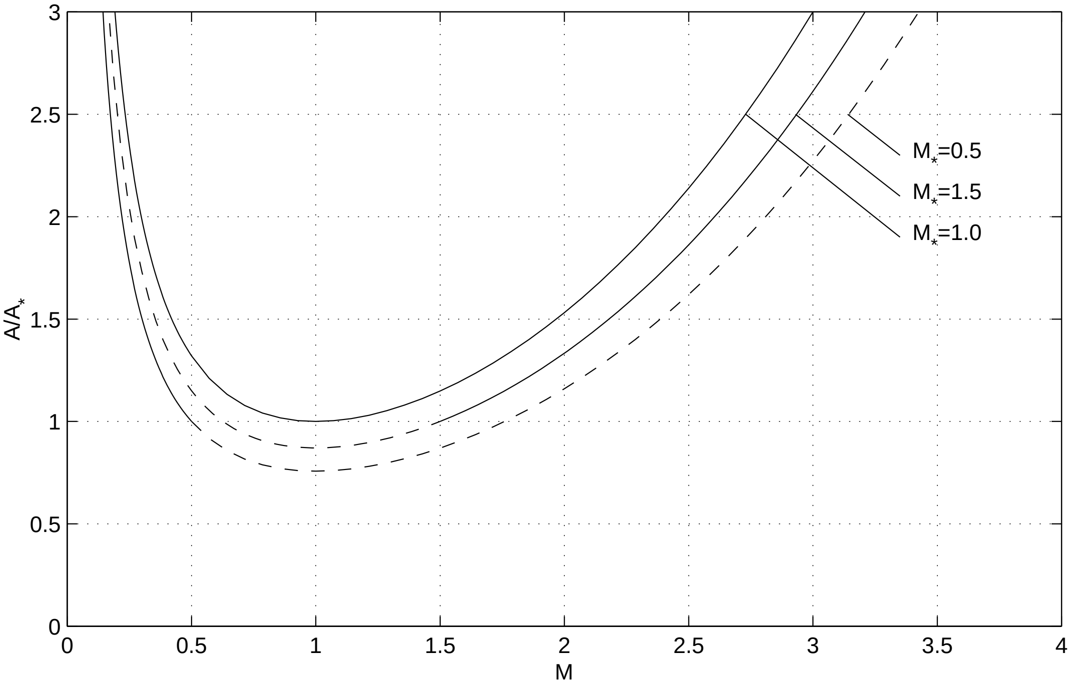

Figure 1.4: Variation of relative nozzle cross sectional area with Mach number.

Using equations (1.16) and (1.17) we find

$$\frac{A}{A_*} = \frac{M_*}{M}\left(\frac{1+\dfrac{\gamma-1}{2}M^2}{1+\dfrac{\gamma-1}{2}M_*^2}\right)^{\frac{\gamma+1}{2(\gamma-1)}}. \tag{1.19}$$

This equation characterizes the Mach number of the flow from the geometry of the nozzle. Plots of equation (1.19) for different values of M_* are shown in Figure 1.4. Notice that the minimum always occurs at $M = 1$ and since $A/A_* \geq 1$ only the $M_* = 1$ curve allows subsonic and supersonic speeds to simultaneously occur in a given nozzle.

1.5 Effect of the Nozzle Geometry

From the results of the shock tube simulation with an initial pressure of $P_L = 4\,\mathrm{atm}$ we found that at the input of the nozzle

$$M_{\mathrm{in}}^2 = 0.23$$

and for an accelerating nozzle we need $M_* = 1$. Therefore equation (1.19) yields

$$\frac{A_{\mathrm{in}}}{A_*} = \frac{1}{M_{\mathrm{in}}}\left(\frac{1+\dfrac{\gamma-1}{2}M_{\mathrm{in}}^2}{1+\dfrac{\gamma-1}{2}}\right)^{\frac{\gamma+1}{2(\gamma-1)}} = 1.36.$$

In a similar fashion the ratio A_{out}/A_* determines M_{out}. If M_{out} is too large the flow in the nozzle will suffer a normal shock which will increase the output pressure and correspondingly decrease the speed of the gas particles. The condition for a normal shock is considered in detail in Section 1.6. For convenience we simulate a nozzle

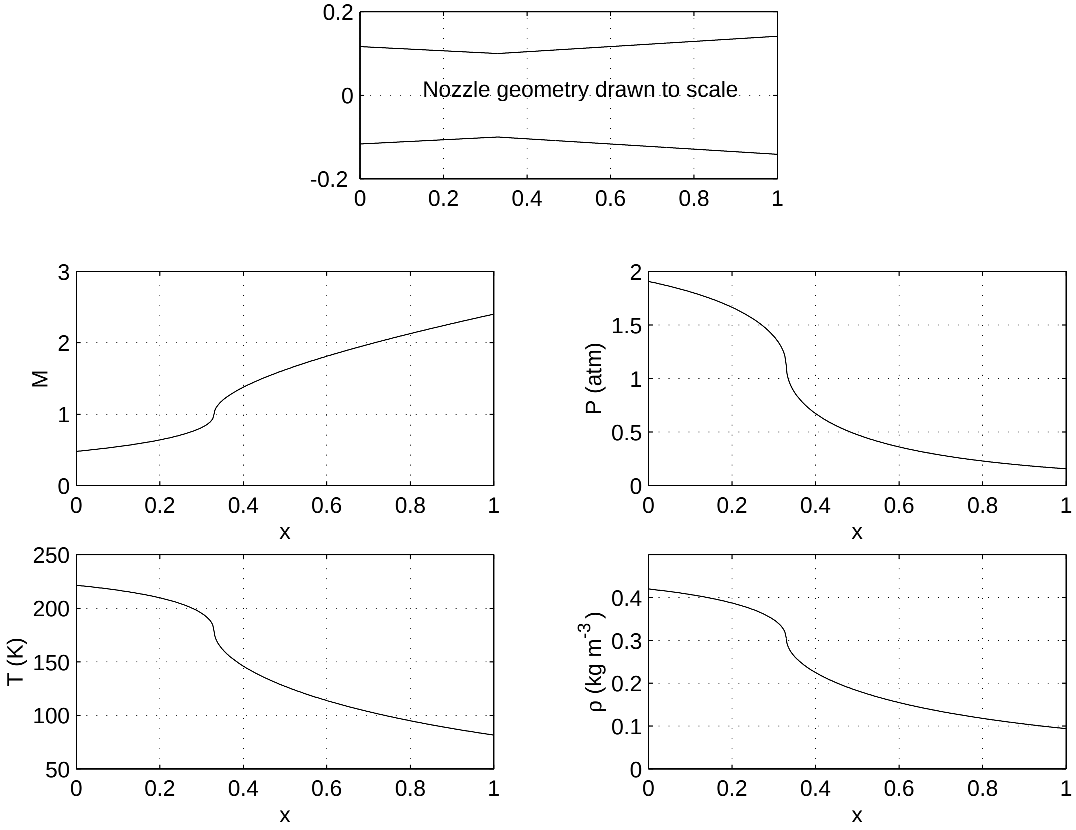

Figure 1.5: Variation of gas properties within a given nozzle.

with $A_{\text{out}}/A_* = 2$ so that $M_{\text{out}} = 2.4$. The plots in Figure 1.5 are obtained directly from expressions (1.16) through (1.19) given the input from the shock tube solution presented in Section 1.4. Along the abscissa of each plot the position is given in terms of the normalized position along the nozzle.

1.6 Normal Shocks

If the Mach number in the diverging section of the nozzle becomes too large then a shock can be produced. Consider a portion of the divergent section of a nozzle with a normal shock with isentropic flow to either side (refer to Figure 1.6). If the shock is at a fixed position then continuity of the flow gives the condition $\rho_1 A v_1 = \rho_2 A v_2$. As one traverses the shock the entropy should increase however both the energy and momentum should be conserved. This results in two additional relationships. The steady state of expression (1.2) yields

$$P_1 + \rho_1 v_1^2 = P_2 + \rho_2 v_2^2$$

for the momentum while a restatement of expression (1.15) gives the energy conservation

$$\frac{v_1^2}{2} + \frac{\gamma}{\gamma - 1}\frac{P_1}{\rho_1} = \frac{v_2^2}{2} + \frac{\gamma}{\gamma - 1}\frac{P_2}{\rho_2}.$$

Systematically eliminating ρ_2, and v_2 from the above equations yields two solutions. $P_2 = P_1$ and

$$P_2 = \frac{1}{\gamma + 1}\left[2\rho_1 v_1^2 + (\gamma - 1)P_1\right].$$

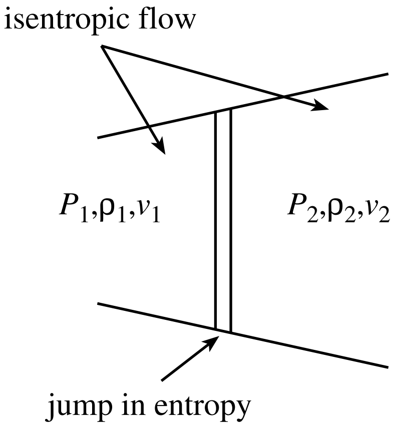

Figure 1.6: A normal shock in the divergent nozzle section.

The first condition implies that there is no shock. For the second condition notice that $\rho_1 v_1^2 = \gamma P_1/c_1^2$ so that simplifying one obtains the shock pressure (dropping the subscripts)

$$P_{\mathrm{S}} \equiv P\left(\frac{2\gamma}{\gamma+1}M^2 - \frac{\gamma-1}{\gamma+1}\right).$$

To avoid a shock we need to ensure that the shock pressure does not fall below the back pressure of the nozzle of $1\,\mathrm{atm}$. It is straightforward to verify that over the divergent section of the nozzle P_{S} has a unique positive maximum and decreases monotonically beyond this point. Checking P_{S} at the nozzle exit we use the values in Section 1.5 ($M_{\mathrm{out}} = 2.4$, $P_{\mathrm{out}} = 0.157\,\mathrm{atm}$) to find that $P_{\mathrm{S}} = 1.09\,\mathrm{atm}$ indicating that there is no shock.

1.7 Drag Force

Having established the flow of the He gas throughout the gene gun we can model the behaviour of the gold micro-particles. The ultimate goal is to maximize the acceleration of these particles for a given initial shock tube pressure P_{L}. We assume that the acceleration is solely due to the drag force of the gas particles on the gold. As a result the equation of motion for the micro-particles can be written as

$$m_{\mathrm{p}}\frac{\mathrm{d}v_{\mathrm{p}}}{\mathrm{d}t} = \frac{1}{2}C_{\mathrm{D}}A_{\mathrm{p}}\rho(v - v_{\mathrm{p}})^2. \tag{1.20}$$

Quantities m_{p}, $v_{\mathrm{p}}$and A_{p} are the mass, speed and cross sectional area of the gold particles, C_{D} is the drag coefficient and v is the speed of the gas as determined in the previous sections. For spherical particles in a flow characterized by a Reynolds number Re $\gtrsim$ 30 the drag coefficient $C_{\mathrm{D}} \simeq 1$ [11]. In addition, typical values for the gold micro-particles used in this manner are [8] $r_{\mathrm{p}} = 3 \times 10^{-6}\,\mathrm{m}$, $\rho_{\mathrm{p}} = 16800\,\mathrm{kg}\,\mathrm{m}^{-3}$ and the combination

$$C_{\mathrm{D}}\frac{A_{\mathrm{p}}}{2m_{\mathrm{p}}} = \Gamma = \frac{3\pi r_{\mathrm{p}}^2}{8\pi r_{\mathrm{p}}^3\rho_{\mathrm{p}}} = \frac{3}{8\rho_{\mathrm{p}}r_{\mathrm{p}}} \simeq 7.44\,\mathrm{m}^2\,\mathrm{kg}^{-1}.$$

The drag force (1.20) has the form

$$f(M) = \rho c^2\left(M - \frac{v_{\mathrm{p}}}{c}\right)^2$$

where ρ and c are functions of the Mach number M and the particle velocity v_{p} is treated as a parameter. Taking the derivative and simplifying we find that

$$\frac{\mathrm{d}f}{\mathrm{d}M} = -\rho c^2\frac{M - \frac{v_{\mathrm{p}}}{c}}{1 + \frac{\gamma-1}{2}M^2}\left(M^2 - \frac{v_{\mathrm{p}}}{c}M - 2\right). \tag{1.21}$$

Since the velocity of the gas is greater than the micro-particles, $v > v_{\mathrm{p}}$ or $M > v_{\mathrm{p}}/c$ indicating that (1.21) is positive provided the speed of the gas particles lies in the range

$$v_{\mathrm{p}} < v < \sqrt{2}\left(c^2 + \frac{v_{\mathrm{p}}^2}{8}\right)^{1/2} + \frac{v_{\mathrm{p}}}{2}.$$

When $v = v_{\mathrm{p}}$ the drag force $f = 0$ so we can conclude that the drag force is maximized if the speed of the gas satisfies

$$v_{\mathrm{opt}} = \sqrt{2}\left(c^2 + \frac{v_{\mathrm{p}}^2}{8}\right)^{1/2} + \frac{v_{\mathrm{p}}}{2}.$$

In the special case when $v_{\mathrm{p}} = 0$ we see that the drag force is maximized when the Mach number of the gas is $M_{\mathrm{opt}} = \sqrt{2}$. This is clearly not attainable unless we place the micro-particles after the throat of the nozzle.

To optimize the drag we break the flow into three separate regimes: approaching the nozzle, reaching optimal drag and tracking optimal drag.

Approaching Nozzle: $(0 \le t \le t_1)$ Before entering the nozzle the gas flow parameters are constant. For the initial pressure pulse of $P_{\mathrm{L}} = 4\,\mathrm{atm}$ the simulation found $M = 0.48$, $\rho = 0.42\,\mathrm{kg\,m^{-3}}$, and $c = 874\,\mathrm{m\,s^{-1}}$. In this regime the speed of the micro-particles evolves according to

$$\frac{\mathrm{d}v_{\mathrm{p}}}{\mathrm{d}t} = \Gamma\rho c^2\left(M - \frac{v_{\mathrm{p}}}{c}\right)^2, \qquad v_{\mathrm{p}}(0) = 0.$$

The position of the particles is

$$x_{\mathrm{p}}(t) = \int_0^t v_{\mathrm{p}}(\tau)\,d\tau$$

and the time at which they enter the nozzle satisfies $x_{\mathrm{p}}(t_1) = L_1$. If the original distance from the gold particles to the nozzle entrance, L_1, is sufficiently large then the micro-particles approach the maximal speed of $Mc = 420\,\mathrm{m\,s^{-1}}$. For example, if $L_1 = 10\,\mathrm{cm}$ then after $t_1 = 0.77\,\mathrm{ms}$ the micro-particles reach the nozzle entrance and are travelling at $v_{\mathrm{p}}(t_1) = v_{\mathrm{p1}} = 267\,\mathrm{m\,s^{-1}}$.

Reaching Optimal Drag: $(t_1 < t \le t_2)$ As the gas passes through the nozzle the Mach number increases monotonically causing the density and the speed of sound to decrease. To maximize the drag force the gold particles need to reach the optimal curve as fast as possible. In this regime the increase in the Mach number is governed by the area variation in the nozzle whereas the increase in the micro-particle velocity is primarily driven by their size and mass.

The time t_2 is defined as the time when

$$M = \sqrt{2}\left(1 + \frac{v_{\mathrm{p}}^2}{8c^2}\right)^{1/2} + \frac{v_{\mathrm{p}}}{2c}$$

which must occur in the diverging section of the nozzle. Recall that $c = c(M)$. In this regime the gold particle velocity evolves according to

$$\frac{\mathrm{d}v_{\mathrm{p}}}{\mathrm{d}t} = \Gamma\rho c^2\left(M - \frac{v_{\mathrm{p}}}{c}\right)^2, \qquad v_{\mathrm{p}}(t_1) = v_{\mathrm{p1}}$$

where ρ and c are now functions of the Mach number and therefore functions of the area ratio A/A_* within the nozzle.

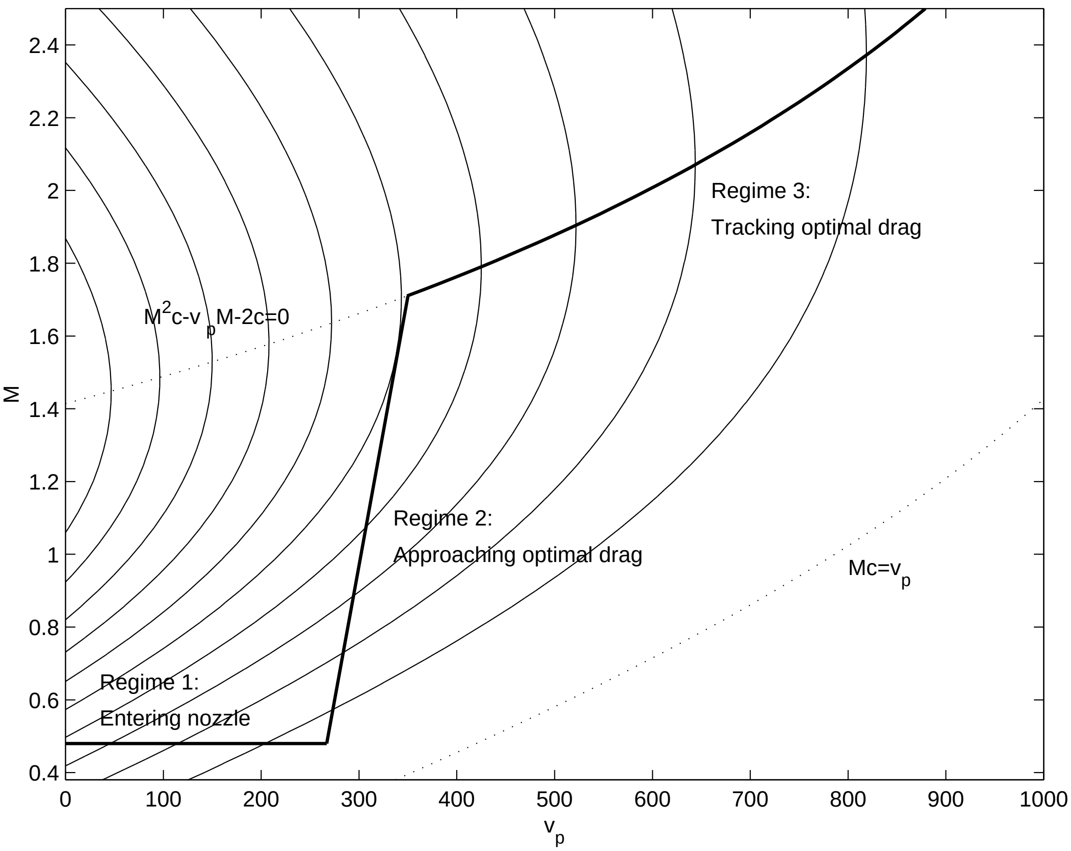

Figure 1.7: Contours of constant drag.

Tracking Optimal Drag: $(t > t_2)$ In diverging section of the nozzle Mach number continues to increase. We have to design the nozzle so that the Mach number evolves according to

$$M^2 - \frac{v_\mathrm{p}}{c} M - 2 = 0$$

where v_p evolves as in the previous regimes. Once the spatial evolution of the Mach number is determined the required area variation is found by solving (1.19). Figure 1.7 illustrates the contours of constant drag, curves of minimal and maximal drag, and a chosen traversal scheme.

1.8 Conclusion and Future Work

The dynamics of the gas and the micro-particles has been determined in the case of a one-dimensional isentropic flow. Additional analysis is required for the second regime of the gas flow, reaching the optimal drag curve. One obvious next step is a comparison of the results predicted in this model with an averaged axisymmetric model.

Another challenge is to model the behaviour of the target. In particular, to reproduce observed penetration depth distributions for a given nozzle geometry. In particular, can the observed variation be solely accounted for by the variation in the initial position and diameter of the micro-particles?

Finally, the ramifications of using a CO_2 cartridge rather than He gas still need to be determined.

Bibliography

[1] Alkhimov, A.P., Kosarev, V.F. & Klinkov, S.V. (2001). *The Features of Cold Spray Nozzle Design.* Journal of Thermal Spray Technology, 10(2), 375–381.

[2] Anderson, J. D. (1984). Modern Compressible Flow. McGraw–Hill: New York.

[3] Dykhuizen, R.C. & Smith, M.F. (1998). *Gas Dynamic Principles of Cold Spray.* Journal of Thermal Spray Technology, 7(2), 205–213.

[4] Grujicic, M., Zhoa, C.L., Tong, C., DeRosset, W.S. & Hefritch, D. (2004). *Analysis of the impact velocity of powder particles in the cold-gas dynamic-spray process.* Materials Science & Engineering A368, 222–230.

[5] Jodoin, B. (2001). *Cold Spray Nozzle Mach Number Limitation.* Journal of Thermal Spray Technology, 11(4), 496–508.

[6] LeVeque, R. J. (1990). Numerical Methods for Conservation Laws. Birkhauser–Verlag: Basel.

[7] LeVeque, R. J. (2002). Finite Volume Methods for Hyperbolic Problems. Cambridge University Press.

[8] Mitchell, T.J., Kendall, M.A.F. & Bellhouse, B.J. (2003). *A ballistic study of micro-particle penetration to the oral mucosa.* International Journal of Impact Engineering, 28, 581–599.

[9] Nabulsi, S., Page, N.W., Duval, A.L., Seabrook, Y.A. & Scott, K.J. (1994). *A gas-driven gene gun for microprojectile methods of genetic engineering.* Measurement Science and Technology, 5, 267–274.

[10] Sod, G. A. (1978). *A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws.* Journal of Computational Physics, 27, 1–31.

[11] Streeter, V.I. (1961). Handbook of Fluid Dynamics. McGraw–Hill: New York.

[12] Thomas, J-L., Bardou, J., L'hoste, S., Mauchamp, B. & Chavancy, G. (2001). *A helium burst biolistic device adapted to penetrate fragile insect tissues.* Journal of Insect Science, available online insectscience.org/1.9.

[13] Thomas, J.W. (1995). Numerical Partial Differential Equations: Finite Difference Methods. In Texts in Applied Mathematics 22, Springer: New York.

[14] Thomas, J.W. (1995). Numerical Partial Differential Equations: Conservation Laws and Elliptic Equations. In Texts in Applied Mathematics 33, Springer: New York.

Chapter 2

Modelling the Temperature Distribute in Concrete Structure

Participants: Tim Myers (Mentor), Boyan Bejanov, Andreas Hofinger, Heejeong Lee, Joohee Lee, Malcolm Roberts, Yan Wu.

PROBLEM STATEMENT: The combination of cement and water produces an exothermic chemical reaction. In industrial processes which involve the creation of large blocks of concrete the heat from the reaction can have a significant effect on the structural properties of the mature concrete. The expansion of the concrete due to temperature changes can induce mechanical stress, which will change the resulting mechanical properties of the concrete slab. Therefore, it is important to determine how heat is distributed in maturing concrete and how this heat can escape.

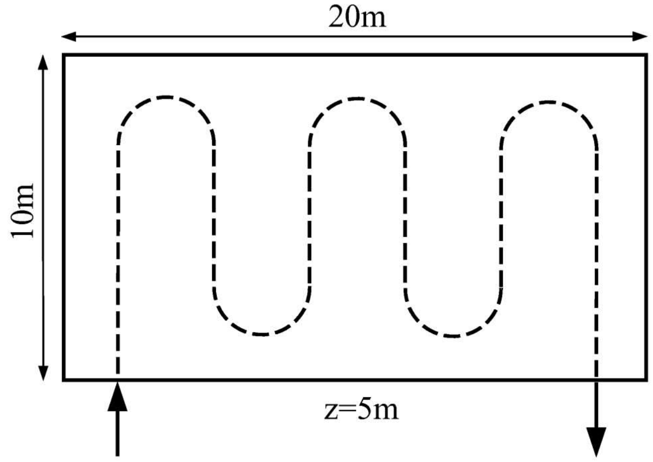

Figure 2.1: Pipe arranged in the concrete block.

2.1 Introduction

The combination of cement and water produces an exothermic chemical reaction. In industrial processes which involve the creation of large blocks of concrete, the heat from the reaction can have a significant effect on the structural properties of the mature concrete, as the expansion of the concrete due to temperature changes can induce mechanical stress, which will change the resulting mechanical properties of the concrete slab. Therefore, it is important to determine how heat is distributed in maturing concrete and how this energy can escape.

One method through which the concrete loses heat is by interaction with the ambient air temperature at the top surface of the concrete. Currently, this is modelled as though the temperature of the surface of the concrete is equal to the ambient temperature. However, this is found to disagree with observations. We were asked by the industrial representative to explain this difference.

When the concrete is first poured, one can also introduce a system of pipes through which water can be forced while the concrete matures, thus cooling the concrete internally. The question arises as to the effect of the pipe length, water temperature, flow, etc. on the temperature of the concrete, and what recommendations can be made to improve heat transfer out of the concrete in this method. Figure 2.1 shows how the pipe is arranged in the block.

2.2 Ambient Temperature

The ambient temperature problem consists of determining the correct model of heat transfer between the concrete block and the surrounding environment.

Our setup for this problem is shown on Figure 2.2. The temperature distribution inside the concrete is governed by the heat equation. As long as we are concerned with the interaction between concrete and air, which occurs only in z direction, we can assume that T depends only on z and t. Then

$$\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial z^2}, \quad t > 0, z \in (0, L), \tag{2.1}$$

where κ is the thermal diffusivity of concrete. At $z = 0$ we impose the boundary condition

$$T|_{z=0} = T_g,$$

where T_g is the temperature of the ground, which can easily be measured and for our purposes is considered known. If we impose the same type of boundary condition at the interface with air (at $z = L$) we will get a

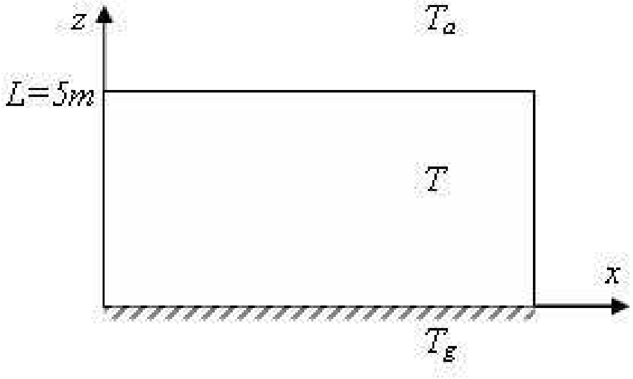

Figure 2.2: The ambient temperature problem.

solution that contradicts the observation. Instead we need to use a Robin type boundary condition

$$k\frac{\partial T}{\partial z}\bigg|_{z=L} = -h\left(T - T_a\right)\big|_{z=L}\,, \tag{2.2}$$

which accounts for the thermal interaction between solid and fluid. In (2.2) k is the thermal conductivity of concrete and $h > 0$ is a physical parameter depending on the material of the solid (concrete), the type of fluid (air), the presence and strength of winds, etc.

The air temperature T_a is assumed to be 24 hrs periodic and of the form

$$T_a(t) = T_{avg} + \Delta T \sin\left(2\pi\frac{t}{24*60*60s}\right).$$

T_{avg} and ΔT can be estimated from meteorological data from the local weather station.

Choosing characteristic values as follows

$$\begin{aligned} \tilde{t} &= \frac{t}{t_c}, \qquad t_c = 24*60*60 \text{ sec.}, \\ \tilde{z} &= \frac{z}{L}, \qquad L = 5 \text{ m}, \\ \tilde{T} &= \frac{T - T_{avg}}{\Delta T}, \end{aligned}$$

we arrive at the non-dimensional problem

$$\begin{aligned} &\frac{\partial \tilde{T}}{\partial \tilde{t}} = K\frac{\partial^2 \tilde{T}}{\partial \tilde{z}^2}, \quad \tilde{t} > 0, \tilde{z} \in (0,1) \\ &\tilde{T}(0,\tilde{z}) = 0 \\ &\tilde{T}(\tilde{t},0) = \tilde{T}_g \\ &\frac{\partial \tilde{T}}{\partial \tilde{z}}\bigg|_{\tilde{z}=1} = -H\left(\tilde{T}(\tilde{t},1) - \sin(2\pi\tilde{t})\right), \end{aligned}$$

where

$$K = \frac{\kappa t_c}{L^2}, \quad \tilde{T}_g = \frac{T_g - T_{avg}}{\Delta T}, \quad H = \frac{hL}{k}$$

and we have imposed initial condition $T = T_{avg}$.

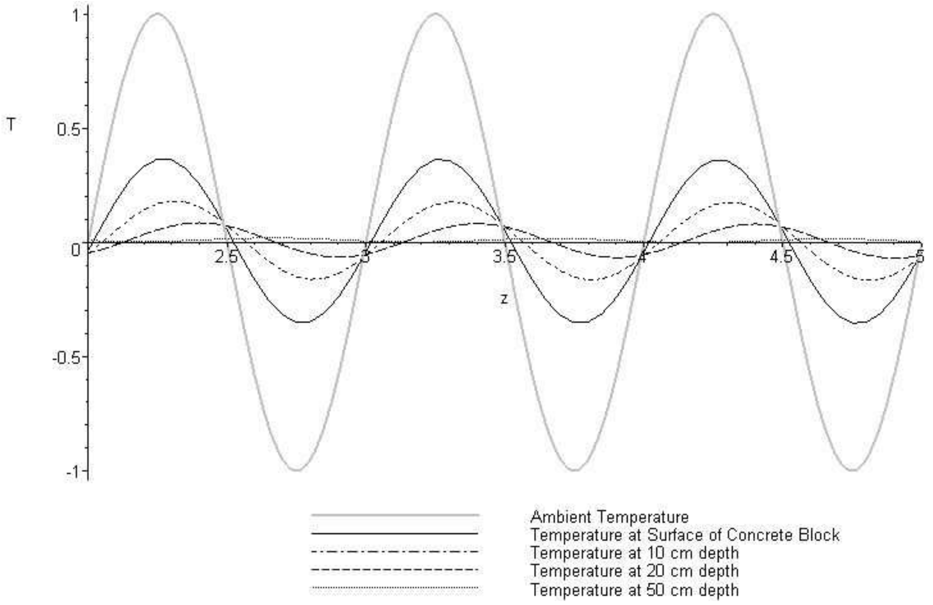

Figure 2.3: Numerical solution to the non-dimensional problem.

The result from a numerical solution of the above IBVP is presented in Figure 2.3. We can see that the desired phase shift is present. Another important observation is that the influence of the outside air on the temperature inside the block diminishes very quickly with depth. We will use this fact in the following section to justify why we can safely ignore this phenomenon when modelling the processes deep inside the concrete.

It also allows us to assume that the concrete block is ”infinitely deep”, i.e. the heat transfer at the top surface will never be felt close to the ground, and redefine our problem on the positive half-line.

$$\begin{aligned}
&\frac{\partial \tilde{T}}{\partial \tilde{t}} = K\frac{\partial^2 \tilde{T}}{\partial \tilde{z}^2}, \quad \tilde{t} > 0, \tilde{z} > 0 \\
&\tilde{T}(0,\tilde{z}) = 0 \\
&\left.\frac{\partial \tilde{T}}{\partial \tilde{z}}\right|_{\tilde{z}=0} = H\left(\tilde{T}(\tilde{t},0) - \sin(2\pi\tilde{t})\right).
\end{aligned}$$

Notice that we have switched the direction of the $\tilde{z}$ axis and now the concrete block extends in $\tilde{z}$ from 0 to $+\infty$. For this problem we can write the solution explicitly

$$\tilde{T} = \frac{2H}{\sqrt{\pi}}\int_0^\infty e^{-H\eta}\int_{\frac{x+\eta}{2\sqrt{Kt}}}^\infty \sin\left[2\pi\left(t - \frac{(x+\eta)^2}{2K\mu^2}\right)\right]e^{-\mu^2}d\mu d\eta.$$

2.3 Influence of the Pipes

In order to investigate the effect of the pipe on the concrete, we consider a cylinder of concrete with pipe running along its axis. We ignore the influence of the outside air because, as shown in the previous section, the effect of ambient temperature does not penetrate very far into the concrete.

Hence, we consider the axisymmetric inhomogeneous heat equation in cylindrical coordinates

$$\frac{\partial T}{\partial t} = \kappa\Delta T + q(t), \tag{2.3}$$

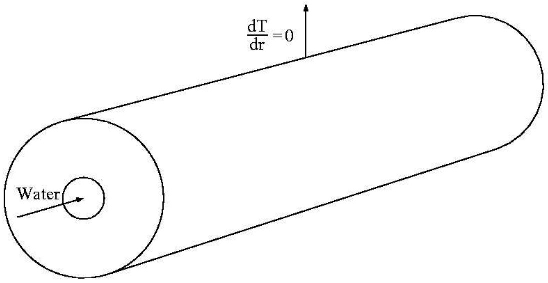

Figure 2.4: Cylinder concrete block with a water pipe at $z = 0$.

where the source term q is due to the heat released by the chemical reaction. Communication with the the industrial representatives suggests that the strength of the chemical reaction diminishes significantly after about 40 hours. Therefore,

$$q(t) = \begin{cases} 40 & 0 \leq t \leq 40 \\ 0 & t > 40. \end{cases} \tag{2.4}$$

We non-dimensionalize this equation using the following characteristic parameters

$$\begin{aligned} \tilde{t} &= \frac{t}{t_c}, \qquad t_c = 24 * 60 * 60 \text{ sec.}, \\ \tilde{z} &= \frac{z}{z_c}, \qquad z_c = 10 \text{ m}, \\ \tilde{r} &= \frac{r}{r_c}, \qquad r_c = 0.5 \text{ m}, \\ \tilde{T} &= \frac{T}{T_c}, \qquad T_c = 40\ ^o\text{C}, \end{aligned}$$

and arrive at the non-dimensional equation

$$\frac{\partial \tilde{T}}{\partial \tilde{t}} = \frac{\kappa t_c}{z_c^2} \frac{\partial^2 \tilde{T}}{\partial \tilde{z}^2} + \frac{\kappa t_c}{r_c^2} \frac{1}{\tilde{r}} \frac{\partial}{\partial \tilde{r}} \left(\tilde{r} \frac{\partial \tilde{T}}{\partial \tilde{r}} \right) + T_c\, \tilde{q}. \tag{2.5}$$

Since $1/z_c^2 << 1/r_c^2$ the diffusion in $\tilde{z}$ is negligible compared to diffusion in $\tilde{r}$. Thus

$$\frac{\partial \tilde{T}}{\partial \tilde{t}} = \frac{\kappa t_c}{r_c^2} \frac{1}{\tilde{r}} \frac{\partial}{\partial \tilde{r}} \left(\tilde{r} \frac{\partial \tilde{T}}{\partial \tilde{r}} \right) + T_c\, \tilde{q}. \tag{2.6}$$

We impose symmetry boundary condition at the outer radius of the cylinder $\tilde{r} = 1$

$$k \frac{\partial \tilde{T}}{\partial \tilde{r}} = 0, \tag{2.7}$$

and we model the interaction with water at $\tilde{r} = \tilde{a} = a/r_c$, where a is the radius of the pipe, with Robin boundary condition

$$\left. \frac{\partial \tilde{T}}{\partial \tilde{r}} \right|_{\tilde{r}=\tilde{a}} = H_p \left(\tilde{T_w} - \tilde{T} \right), \tag{2.8}$$

where $\tilde{T_w} = T_w/T_c$ is the non-dimensionalized temperature of the water and $H_w = h_w r_c/k_w$.

The temperature of the water is governed by the following axisymmetric heat equation

$$\rho_w c_w \frac{\partial T_w}{\partial t} + u\frac{\partial T_w}{\partial z} + v\frac{\partial T_w}{\partial r} = k_w \Delta T_w \tag{2.9}$$

where ρ_w is the density of water, c_w is the specific heat, k_w is the thermal diffusivity, and u and v are the velocity components in z and r direction respectively.

We assume that the water is turbulent and due to the high average speed in z direction and mixing processes in r the water temperature is independent of r and diffusion is negligible. Then

$$\rho_w c_w \frac{\partial T_w}{\partial t} + u\frac{\partial T_w}{\partial z} = 0. \tag{2.10}$$

We now consider the steady state situation. This is of interest, because even though the system may fail to reach the steady state within the 40 hours work of the reaction, the steady state will give us an upper bound of the temperature, that is, the worst case scenario. We set $\frac{\partial \tilde{T}}{\partial \tilde{t}} = 0$ in (2.6), integrate twice and take into account the boundary conditions (2.7) and (2.8) to arrive at the solution

$$\tilde{T} = -\frac{T_c}{K}\tilde{q}\tilde{r} + \frac{T_c}{K}\tilde{q}\ln(\tilde{r}) + \tilde{T}_w + \beta, \tag{2.11}$$

where $K = \kappa t_c / r_c^2$ and the constant β is

$$\beta = \frac{T_c}{K}\tilde{q}\left(\tilde{a} - \ln(\tilde{a}) + \frac{\tilde{a}-1}{H_p \tilde{a}}\right).$$

So we know the temperature of concrete as a function of the temperature of the water, which remains to be determined. We determine the temperature of the water using an energy conservation argument. Since we have argued that heat flows only radially in the concrete, and only along $\tilde{z}$ in the water, it follows that

$$\left.\frac{\partial \tilde{T}}{\partial \tilde{r}}\right|_{\tilde{r}=\tilde{a}} = \lambda \frac{\partial \tilde{T_w}}{\partial \tilde{z}},$$

where λ is a non-dimensional coefficient of proportionality depending on the physical parameters of the problem. We do not specifically find an expression for λ. Instead, we use the boundary conditions (2.8) to obtain

$$\begin{aligned} \lambda \frac{\partial \tilde{T_w}}{\partial \tilde{z}} &= \left.\frac{\partial \tilde{T}}{\partial \tilde{r}}\right|_{\tilde{r}=\tilde{a}} \\ &= H_p\left(\tilde{T_w} + \frac{T_c}{K}\tilde{q}\tilde{r} - \frac{T_c}{K}\tilde{q}\ln(\tilde{r}) - \tilde{T}_w - \beta\right) \\ &= \text{const.} \end{aligned}$$

Therefore, the temperature of the water is linear in z. Since we are considering the steady state we can use a global energy conservation argument. All the thermal energy produced by the chemical reaction is absorbed by the water, which results in temperature increase. If we denote by $T_\triangle$ the change in water temperature along z we can write the following energy balance equation

$$T_\triangle f_w \rho_w c_w = qV\rho c,$$

where ρ_w is the density of the water, f_w is the water flux in m^3/h, ρ and c are the density and the specific heat of concrete and V is the total volume of the concrete cylinder. This equation yields

$$T_\triangle = qV_c c/(f_w c_w).$$

Figure 2.5 shows the temperature profile inside the concrete. The gradient (and therefore also the stress in the concrete) is largest close to the pipe.

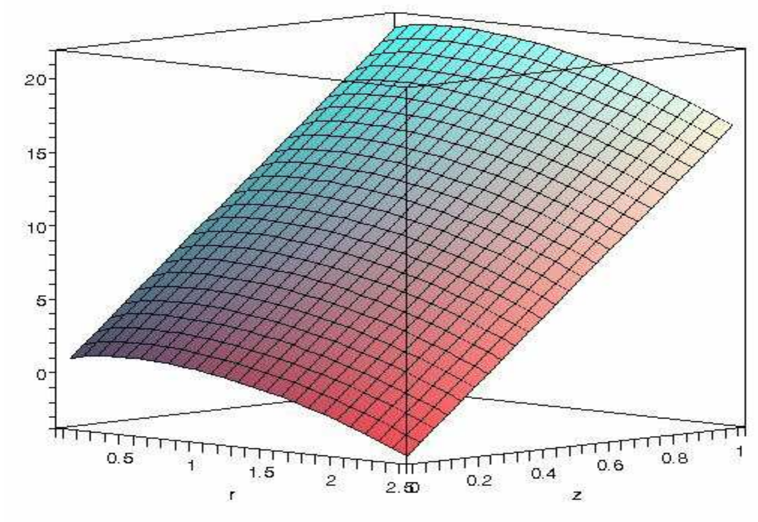

Figure 2.5: Temperature distribution in the steady state for typical parameters.

2.4 Conclusion

Boundary Conditions: The industrial representative who originally presented this problem assumed that the temperature of the surface of the unheated concrete slab was equal to the ambient temperature. This was not supported by experiment. Mixed boundary conditions, representing a limited heat flux through the surface of the slab, resulted in numerical and analytic results which agreed with experiment. Further, using dimensional arguments and simulations, we showed that the effect of the ambient temperature was not a significant factor in cooling the block in comparison to forcing water through pipes put into the block.

Time dependant results: We solved the time dependant system numerically. The system has the largest temperature gradient and maximum temperature in the steady state, which is eventually attained by the system if the chemical reaction is sustained for a long enough time. The chemical reaction in the concrete lasts approximately 48 hours, and the time scale for heat diffusion in the concrete in the radial direction is 3.7 days when the radius of the concrete cylinder is 0.5 m. So, while the system may not attain a steady state, it has time to approach it, and the initial conditions are not totally dominant for the period when the reaction is taking place.

Steady state results and suggestions for industry: The steady state gave us insight into the overall behaviour of the system. From the analytic solutions we derived, we have a number of suggestions to reduce heat stress in concrete.

- The temperature of the concrete is directly related to the input temperature. However, the temperature gradient near the pipe also increases when the input temperature drops.
- The heat conductivity of the water pipe does not significantly effect the cooling of the concrete.
- The temperature of the concrete is inversely proportional to the flow rate of the water.
- More pipes reduces the maximum temperature and reduces the maximum temperature gradient.
- Wider pipes create a smaller temperature gradient than thin pipes.

Thus the best situation would be to pump relatively warm water through a network of large pipes in the concrete. Not only will this reduce the overall temperature, but it will also reduce the stresses in the concrete and the time until the system reaches equilibrium.

Bibliography

[1] Carslaw, H.S. and Jaeger, J. C., *Conduction of Heat in Solids*. Clarendon Press, Oxford, 1986.

Chapter 3

Modelling Nonlinear Pulse Propagation in Optical Transmission Lines

Participants: Tobias Schäfer (Mentor), Benjamin Akers, Amirhossein Amiraslani, Mohammad Al-Khaleel, Wan Chen, Radu Haiduc, Lee Jinwoo, Andrei Maxim, Stanislava Peker, Azar Shakoori, Wang Zhian.

PROBLEM STATEMENT: We are considering the problem of sending data in the form of light pulses through optical fibres. To do so, we assume data is transmitted as sequence of $1's$ and $0's$ where a pulse corresponds to a one and a zero amplitude of the signal corresponds to a zero of the bit stream. Due to the interaction between the light and the material of the fibre line, the amplitude of light pulse at the end of the transmission might become so small that information will be lost. The main issue here is how to fix this problem and make pulses last as long as possible in order to be able to transmit data over large distances. The mathematical model of a pulse travelling in an optical fibre is the cubic nonlinear Schrödinger equation. We study the solutions of this equation for different parameter regimes trying to achieve stable pulse propagation.

3.1 Introduction

The possibility of sending tremendous amounts of data in the form of light pulses through optical fibres has transformed the world in the recent years. High-speed internet and video on demand are only two examples of exciting new communication technologies. When using a fibre line for such data transmission it natural to try to send data without information loss, meaning that the signal at the output should be the same as the signal at the input. But several effects change the signal on its way through the fibre. As we already mentioned in the problem statement, the amplitude of the light pulse will decrease during the transmission. The receiver reads any amplitude smaller than a threshold T as a 0 and any amplitude greater than T as a 1. Figure 3.1 shows this decay of the amplitude in a numerical simulation. The decrease of amplitude is not only due to loss of the signal's energy in the fibre. The effect of loss can be taken care of by boosting the signal with amplifiers. The more serious effect is *dispersion* which leads to spreading of the pulse and thus to decay of the amplitude. Consider two neighbouring pulses which will spread out and start to overlap. At a certain moment, it will not be possible anymore to decode the pulses as two ones, even after amplification.

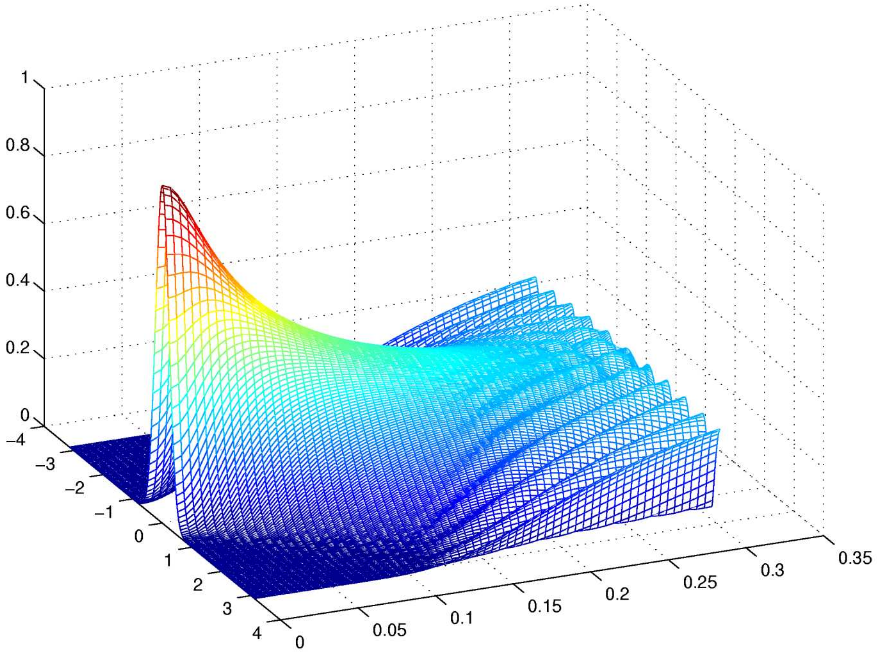

Figure 3.1: Decrease of the pulse amplitude during propagation. One unit of the propagation corresponds to a distance of 10 km.

We can view approach problem from more mathematical point of view by first considering the fact that pulse propagation is modelled by Maxwell's equations. By reducing Maxwell's equations making use of the fact that the pulse widths are larger than the oscillation periods of the carrier wave, we come up with a well-known physical equation called the **Schrödinger** equation, shortly **NLSE** [3].

$$\frac{\partial A}{\partial Z} + \frac{\alpha}{2}A + \frac{i\beta_2}{2}\frac{\partial^2 A}{\partial t^2} = i\gamma|A|^2 A. \tag{3.1}$$

Here, A is the slowly varying envelope of the electric field, and α, β_2, γ are parameters provided by the manufacturer of the fibre line. Typical values of the parameters are as follows: $\alpha = -0.05km^{-1}, \gamma = 1km^{-1}, \beta_2 = 20ps^2km^{-1}$. To build a mathematical model, it is useful to make equation (3.1) dimensionless. To do so, we may scale time t by a characteristic time corresponding to a typical pulse width T_{FWHM} in the range of several picoseconds, power through the peak power P_{Peak} (usually several mW) and space z by a characteristic length L of several tens of kilometres. In addition, we make a phase transform in order to incorporate the lossy term in equation (3.1) into the nonlinear coefficient. Let us see how these transforms change the structure of (3.1). The scales can be written as coordinate transforms, hence

$$\tilde{A} = A/\sqrt{P_{\text{Peak}}}, \qquad \tilde{z} = z/L, \qquad \tilde{t} = t/T_{\text{FWHM}}. \tag{3.2}$$

In those new variables, we find the equation for $\tilde{A}$

$$\tilde{A}_{\tilde{z}} + \frac{\alpha L}{2}\tilde{A} + i\frac{\beta_2 L}{2T_{\text{FWHM}}^2}\tilde{A}_{\tilde{t}\tilde{t}} = i\gamma P_{\text{Peak}} L|\tilde{A}|^2\tilde{A}.$$

Introducing a phase transform by

$$\tilde{A}(\tilde{z},\tilde{t}) = \mathcal{A}(\tilde{z},\tilde{t})\mathrm{e}^{-\frac{\alpha}{2}L\tilde{z}} \tag{3.3}$$

we obtain

$$i\mathcal{A}_{\tilde{z}} - \frac{1}{2}\frac{\beta_2 L}{T_{\text{FWHM}}^2}\mathcal{A}_{\tilde{t}\tilde{t}} + \gamma P_{\text{Peak}} L\mathrm{e}^{-\alpha L\tilde{z}}|\mathcal{A}|^2\mathcal{A} = 0.$$

Writing the coefficients as functions of the evolution variable (in general, β_2 and γ can vary with the distance, since the fibre line normally consists of fibres of different type), and by finally renaming all the variables, the dimensionless NLSE becomes

$$iA_z + d(z)A_{tt} + c(z)|A|^2A = 0. \tag{3.4}$$

The variable $z \in [0,\infty)$ corresponds to the propagation distance of the pulse from the beginning of the fibre line. The time variable $t \in (-\infty,\infty)$ presents the window in which the signal (the ones and zeros) are encoded.

To be able to handle the problem we decided to look first at the linear part of the equation (3.4) and then turn to the nonlinear problem.

3.2 Linear Problem

Let us first look at the linear Schrödinger equation:

$$iA_z + d(z)A_{tt} = 0 \quad \text{with initial condition} \quad A(z=0,t) = N\exp{(-\alpha t^2)}. \tag{3.5}$$

To study equation (3.5) with respect to our problem, we analyze it in three steps: First, we compute an explicit analytical solution using Fourier transform. Then we make use of this solution to characterize the dispersion-induced pulse spreading. The third step is to suggest a technique to compensate for this spreading in order to achieve stable pulse propagation.

3.2.1 Analytical Solution

The analytical solution of equation (3.5) can be obtained by Fourier transform assuming vanishing boundary conditions $|A| \to 0$ for $|t| \to \infty$. In the following equations we recall Fourier transform and back transform respectively:

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t)\exp{(i\omega t)}dt, \tag{3.6}$$

$$f(t) = \frac{1}{2\pi}\int \hat{f}(\omega)\exp{(-i\omega t)}dt. \tag{3.7}$$

The linear Schrödinger equation is a partial differential equation (PDE) in the variables z and t, with initial value as function of t, given as follows:

$$iA_z + d(z)A_{tt} = 0, \quad A(0,t) = N\exp{(-\alpha t^2)}. \tag{3.8}$$

After Fourier transform we obtain associated ordinary differential equations (ODEs) for the Fourier modes. The initial values in Fourier domain are found from the initial value in time domain:

$$i\hat{A}_z - \omega^2 d(z)\hat{A} = 0, \quad \hat{A}(0,\omega) = N\sqrt{\frac{\pi}{\alpha}}\exp\left(\frac{-\omega^2}{4\alpha}\right). \tag{3.9}$$

The solution of the ODEs in equation (3.9) in Fourier domain can be found by direct integration:

$$\hat{A}(z,\omega) = N\sqrt{\frac{\pi}{\alpha}}\exp\left(\frac{-\omega^2}{4\alpha}\right)\exp\left(-i\omega^2 R(z)\right) \tag{3.10}$$

where $R(z)$ is the so-called *accumulated dispersion*,

$$R(z) = \int_0^z d(\xi)d\xi. \tag{3.11}$$

Using back Fourier transform in equation (3.7), the solution of (3.9) in time domain is:

$$A(z,t) = \frac{N\exp\left(\frac{-\alpha t^2}{4\alpha R(z)i+1}\right)}{\sqrt{4\alpha R(z)i+1}}. \tag{3.12}$$

3.2.2 Dispersive Spreading

As we saw in Figure 3.1, in the real world we will lose some information during the transmission. Since we computed the analytical solution to the linear problem, we are now able to make predictions how far we can propagate without information loss.

$$|A| = \frac{N}{\sqrt[4]{1+16\alpha^2R^2(z)}}\exp\left(\frac{-\alpha t^2}{1+16\alpha^2R^2(z)}\right). \tag{3.13}$$

If we want at $t = 0$, the amplitude be half of the amplitude of the original signal, hence $|A| = \frac{N}{2}$ in order to estimate the pulse spreading as a function of the distance, we find:

$$\frac{N}{\sqrt[4]{1+16\alpha^2R^2(z)}} = \frac{N}{2}. \tag{3.14}$$

We can solve this equation for the accumulated dispersion $R(z)$.

$$R(z) = \frac{\sqrt{15}}{4\alpha}. \tag{3.15}$$

This means we must find z such that

$$\int_0^z d(\xi)d\xi = \frac{\sqrt{15}}{4\alpha}. \tag{3.16}$$

In practice, the case of one fibre type with constant dispersion is of particular interest. Set $d(z) = d$ and we find from the above relation:

$$z = \frac{\sqrt{15}}{4\alpha d}. \tag{3.17}$$

What does this result mean in terms of real-world quantities? Let's consider for example the line with $40Gbit/s$. This means that each bit will occupy a time window or bit slot of $25ps$. A typical with pulse width is then $\sim 5ps$. If we choose a line with a standard dispersion coefficient as $\beta_2 = -20ps^2/km$, we find that we can propagate about $3km$. This, of course, is only an estimation, but we see that this is by far not satisfactory for fibre lines between cities, or even between continents. Therefore, we have to find a solution to prevent the pulse from dispersive spreading.

3.2.3 Linear Dispersion Management

Recall that in the above solution of the linear problem we have introduced a new function $R(z)$ and called it the accumulated dispersion. In fact, the solution of the linear problem only depends on the initial condition and on the value of $R(z)$. If we evaluate $R(z)$ at the end of the fibre line, let's say after a distance L, we see that, provided $R(L) = 0$, we will have complete recovery of our initial signal $A(0,t)$, hence $A(0,t) = A(L,t)$:

$$\begin{aligned} iA_z + d(z)A_{tt} &= 0 \\ i\hat{A}_z - d(z)\omega^2\hat{A} &= 0 \\ \hat{A}(z,\omega) &= \exp\left(-iR(z)\omega^2\right)\hat{A}(0,\omega) \\ R(z) &= \int_0^z d(\xi)d\xi \\ R(L) = \int_0^z d(\xi)d\xi &= 0 \Longrightarrow \hat{A}(L,\omega) = \hat{A}(0,\omega). \end{aligned}$$

The condition $R(L) = 0$ can be achieved by using more than one fibre type, hence building a fibre system by a concatenation of fibres whose dispersion coefficients have alternating signs. This technique is called *dispersion management*. The result of such a management is presented in Figure 3.2.

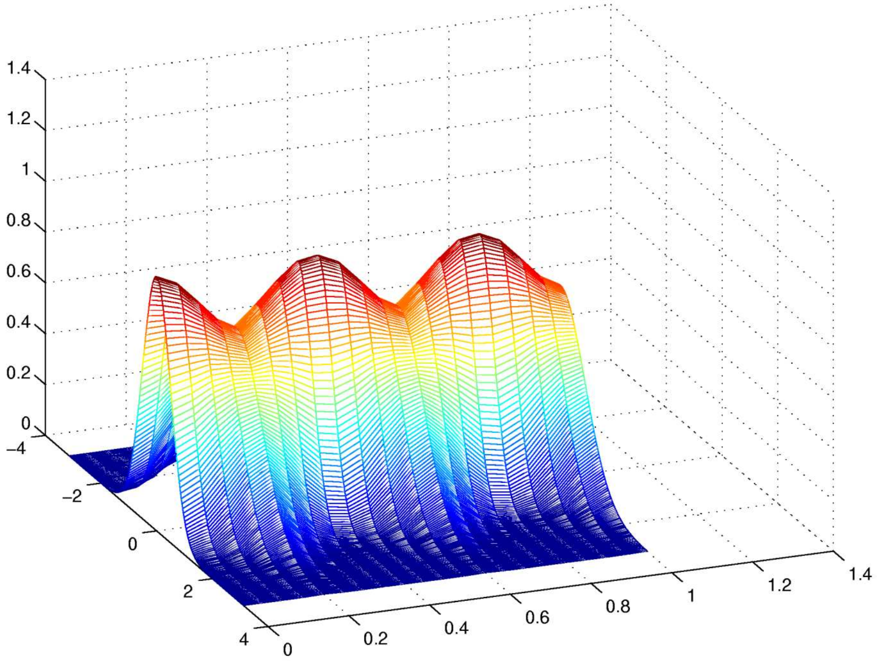

Figure 3.2: Linear dispersion management. The pulse breathes periodically and, at the end of each dispersion period, it recovers the original shape.

3.3 Nonlinear Problem

As a next step, we have to incorporate the nonlinear part of equation (3.1). Here, we consider two scenarios. First, we look at the case of a Schrödinger equation with constant coefficients. This is an appropriate model for systems with one fibre type and so-called distributed amplification. Then, in a second step, we turn to the previously discussed dispersion-managed systems in order to see how nonlinearity will affect pulse propagation in this case.

3.3.1 NLSE with Constant Coefficients

To start, we look at the cubic nonlinear Schrödinger equation with constant coefficients. In this case, we can always rescale in a way to obtain the following form

$$iA_z + A_{tt} + 2|A|^2 A = 0. \tag{3.18}$$

Here, the dispersion is a constant that equals one. The analytical solution of the linear case taught us that, in the absence of nonlinearity, the dispersion term A_{tt} will lead to pulse spreading. Now, we have an additional term, $2|A|^2 A$, and the question is whether it is possible to create a balance between dispersion and nonlinearity in order to obtain a stable pulse whose amplitude will not decrease. Indeed, we can show that

$$A(z,t) = \frac{\lambda}{\cosh(\lambda t)} \exp(i\lambda^2 z)$$

is a solution to (3.18) with the initial condition:

$$A(0,t) = \frac{\lambda}{\cosh(\lambda t)}. \tag{3.19}$$

The fact that this is an analytical solution can be easily checked by substituting the solution into the equation: We have

$$\begin{aligned} \tanh^2(\lambda t) &= 1 - \mathrm{sech}^2(\lambda t) \\ \frac{d}{dt}(\mathrm{sech}(\lambda t)) &= -\lambda \tanh(\lambda t) \end{aligned}$$

Therefore we can find all the derivatives of A as

$$\begin{aligned} A_t &= \lambda(-\tanh(\lambda t))A \\ A_{tt} &= -\lambda^2 \mathrm{sech}^2(\lambda t)A + \lambda^2 \tanh^2(\lambda t)A \\ &= \lambda^2 A(1 - 2\mathrm{sech}^2(\lambda t)) \end{aligned}$$

and for the partial derivative with respect to the evolution variable z we find

$$A_z = -i\lambda^2 A$$

Putting everything together completes our proof:

$$iA_z + A_{tt} + 2|A|^2 A = -\lambda^2 A + \lambda^2 A(1 - 2\mathrm{sech}^2(\lambda t)) + 2\lambda^2 \mathrm{sech}^2(\lambda t)A = 0.$$

This means that $|A(z,t)|$ will always maintain its shape along the whole fibre for arbitrary distances z. Those stable solutions of nonlinear wave equations are called *solitons*. Figure 3.3 shows the propagation of a soliton in a standard fibre line over 100km. What parameters are necessary in order to create a soliton in a practical transmission line? Consider the following typical physical values

$$\begin{aligned} L &= 62.5km \\ T &= 25ps \\ \beta_2 &= -20ps^2/km \\ \gamma &= 1/(Wkm) \end{aligned}$$

From here we find that we need a peak power P_{peak} of $31.25mW$ in order to achieve soliton propagation.

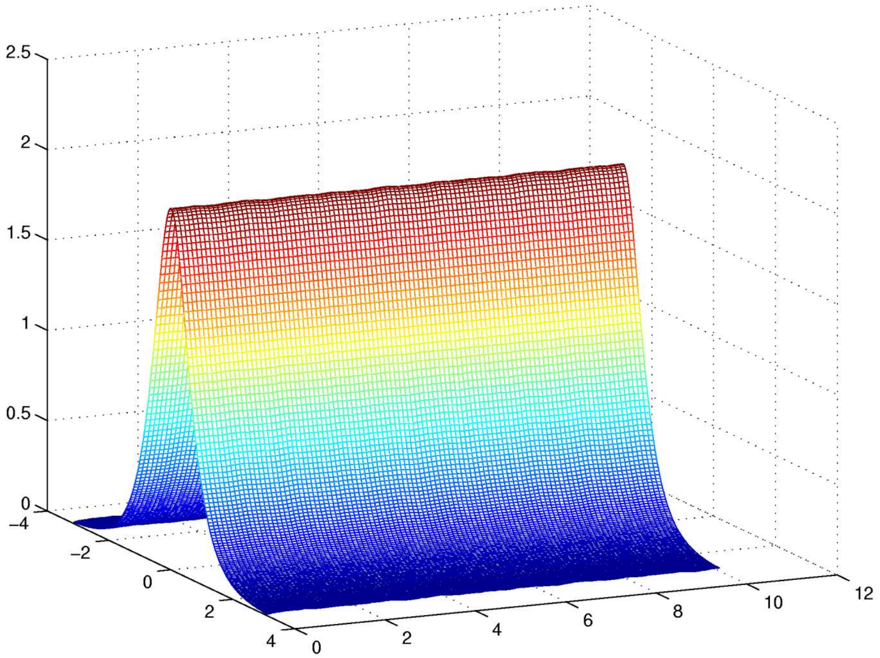

Figure 3.3: Soliton propagation. Nonlinearity balances dispersion and leads to stable pulse propagation. This plot only shows the amplitude. The phase of the signal is rotating with constant speed.

3.3.2 Non-constant Dispersion

What happens to the possibility of a balance between nonlinearity and dispersion in the case of dispersion managed systems? We can think of the dispersion consisting of two different parts: A part $\tilde{d}(z)$ that represents the local variations of the dispersion management and a small, residual part $\langle d \rangle$ that stands for the average constant dispersion. We looked at the equation

$$iA_z + d(z)A_{tt} + \epsilon|A|^2A = 0 \tag{3.20}$$

numerically and found that in certain parameter regimes, it can have stable pulses, so-called ***dispersion-managed solitons***, that are created by a balance between the small nonlinear term and the small residual dispersion. Figure 3.4 shows the evolution of such a dispersion-managed soliton. How can we describe those solutions analytically? Since the coefficients of the above equation are varying, it is difficult to find an analytical solution. But the small parameter gives hope that we can use a perturbation technique in order to find an approximation solution. Here use a multiple scale expansion [2, 1]. We write A with multiple scales in z, hence

$$A(z,t) = A_0(Z_0, Z_1, ...; t) + \epsilon A_1(Z_0, Z_1, ...; t) + ..., \qquad Z_n = \epsilon^n z. \tag{3.21}$$

The function d only depends on the fast variable Z_0. Therefore, we obtain as zero order equation in Fourier domain

$$i\hat{A}_{Z_0} - i\tilde{d}(Z_0)\omega^2\hat{A} = 0 \tag{3.22}$$

and the solution of this equation is

$$\hat{A}_0(Z_0, Z_1, ...; \omega) = \mathrm{e}^{-iR(Z_0)\omega^2}\hat{U}_0(Z_1, Z_2, ...; \omega) \tag{3.23}$$

where the “slow” amplitude $\hat{U}_0$ still depends on all “slow” variables $\{Z_1, Z_2, ...\}$. As usual, the equation for $\hat{U}_0$ will be given by the solvability condition of the first order equation for A, hence the equation for A_1. This

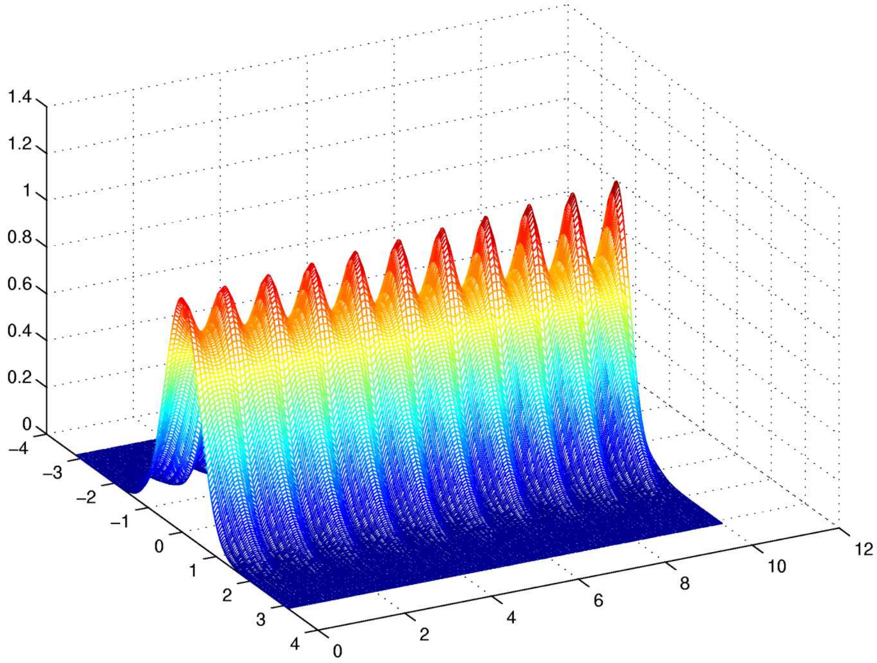

Figure 3.4: Propagation of a stable pulse (dispersion managed soliton) in a system with dispersion management.

equation can be written as

$$iA_{1Z_0} + \tilde{d}(Z_0)A_{1tt} = f(Z_0, Z_1, ...; t), \tag{3.24}$$

where the function f is given by terms involving A_0 that arise from derivatives with respect to the slow variable Z_1, the residual dispersion $\langle d\rangle$ and the nonlinearity, hence

$$f(Z_0, Z_1, ...; t) = -iA_{0Z_1} - \langle d\rangle A_{0tt} - |A_0|^2 A_0. \tag{3.25}$$

In Fourier domain we rewrite (3.24) as

$$i\hat{A}_{1Z_0} - i\tilde{d}(Z_0)\omega^2\hat{A}_1 = \hat{f}(Z_0, Z_1, ...; \omega), \tag{3.26}$$

an equation we can easily solve by variation of the constant: Let

$$\hat{A}_1(Z_0, Z_1, ...; \omega) = \mathrm{e}^{-i\omega^2 R(Z_0)}\hat{C}(Z_0, Z_1, ...; \omega), \tag{3.27}$$

and we obtain

$$\hat{C}(Z_0, Z_1, ...; \omega) = -i\int_0^{Z_0} \hat{f}(\xi, Z_1, ...; \omega)\mathrm{e}^{i\omega^2 R(\xi)}d\xi. \tag{3.28}$$

As a solvability condition we now require that A_1 does not grow, or, stronger

$$\hat{A}_1(1, Z_1, Z_2, ...; \omega) = 0 \tag{3.29}$$

as the two-step map has a period 1. Or, as $R(1) = 0$,

$$\int_0^1 \hat{f}(\xi, Z_1, ...; \omega)\mathrm{e}^{i\omega^2 R(\xi)}d\xi = 0. \tag{3.30}$$

By virtue of (3.25), we can rewrite this as an equation for $\hat{U}(Z_1, Z_2, ...; \omega)$ and obtain (we now write $\hat{U}$ as a function of Z_1 and ω as we do not consider higher orders at this stage),

$$\begin{gathered} -i\hat{U}_{Z_1} + \langle d\rangle\omega^2 - \frac{1}{(2\pi)^2}\int\int\int\int_0^1 \hat{U}(Z_1,\omega_1)\hat{U}(Z_1,\omega_2)\hat{U}(Z_1,\omega_3)\,\cdot \\ \mathrm{e}^{-iR(\xi)(\omega_1^2+\omega_2^2-\omega_3^2-\omega^2)}\delta(\omega_1+\omega_2-\omega_3-\omega)d\xi d\omega_1 d\omega_2 d\omega_3 = 0. \end{gathered} \tag{3.31}$$

This is the so-called *Gabitov-Turitsyn equation*. For the case of dispersion-managed solitons, $\hat{U}$ should be written as

$$\hat{U}(Z_1,\omega) = \mathrm{e}^{ikZ_1}\hat{u}(\omega). \tag{3.32}$$

This is an analogy to the system with constant coefficient where the phase dependence of the solution was a rotation with constant speed. Then we obtain

$$\begin{aligned} (k + \langle d\rangle\omega^2)\hat{u}(\omega) &= \frac{1}{(2\pi)^2}\int\int\int\int_0^1 \hat{u}(\omega_1)\hat{u}(\omega_2)\hat{u}(\omega_3)\,\cdot \\ &\quad \mathrm{e}^{-iR(\xi)(\omega_1^2+\omega_2^2-\omega_3^2-\omega^2)}\delta(\omega_1+\omega_2-\omega_3-\omega)d\xi d\omega_1 d\omega_2 d\omega_3. \end{aligned} \tag{3.33}$$

This nonlocal equation characterizes dispersion-managed solitons. By numerical solution it is possible to find its shape u and parameters where they exist. Again we see that dispersion managed solitons can exist because of a delicate balance between (a nonlocal) nonlinearity and (averaged) dispersion.

3.4 Conclusion

The problem we considered was to find a way to create stable pulses in optical transmission lines. The mathematical model we used was the cubic nonlinear Schrödinger equation (NLSE). For systems operating on very low input power, we can neglect nonlinearity. In a system with constant dispersion, an initially localized pulse will spread and we computed this pulse spreading explicitly. A remedy was introduction of fibres with alternating signs of dispersion (dispersion-management). In the nonlinear case we saw that the nonlinearity can prevent dispersive spreading of special pulses called solitons. We computed the energy necessary for the creation of a soliton in a typical fibre line. This idea of balance between nonlinearity and dispersion can be carried through to dispersion-managed systems leading to the Gabitov-Turitsyn equation which allows to characterize dispersion-managed solitons.

Bibliography

[1] M. J. Ablowitz and G. Biondini. Multiscale pulse dynamics in communication systems with strong dispersion management. *Opt. Lett.*, 23:1668–1670, 1998.

[2] I. Gabitov and S. K. Turitsyn. Averaged pulse dynamics in a cascaded transmission system with passive dispersion compensation. *Opt. Lett.*, 21:327–329, 1996.

[3] A. Hasegawa and Yuki Kodama. *Solitions in optical communications*. Clarendon Press, Oxford, 1995.

Chapter 4

A Game Theoretical Approach to Modelling Network Growth

Participants: Petra Berenbrink (Mentor), Benjamin Chan, Michelle Edwards, Robert Junli Liao, Jason Lobb, Abramov Vilen.

PROBLEM STATEMENT: During the last years, researchers developed a lot of mathematical tools that can help us to understand the von Neuman computer and parallel machines. The Internet is very different to these computers in that it is built, operated and used by a multitude of diverse economic interests in varying relationships of collaboration and competition with each other. This suggests that mathematical tools for understanding the Internet may come from a fusion of algorithmic ideas with concepts from mathematical economics and game theory. We are trying to define and analyse a simple game that can be used to model the growth of the Internet.

4.1 Introduction

Since the United States first brought forth the plan to construct the National Information Infrastructure (NII) in the beginning of 1990s, the Internet has become a huge network consisting of more than 12,000 subnetworks or so-called internet agents. However, the Internet is not designed by any single person or organization. Each subnetwork joins to it by connecting to others with arbitrary links in attempt to generate the most benefits for itself. But from the point of view of the whole network, the corresponding structure may not be the optimal choice. It may cause troubles from path re-route, traffic engineering, load balance, to hot-point overload and so forth.

Because of its unprecedented organization, traditional tools that computer scientists use to investigate von Neuman computer and parallel machines are not suitable for studying the Internet. A completely different tool is required. In [1], the authors proposed to use a game theoretical approach. The Internet can be represented by an abstract graph. Each subnetwork (agent) is a node, it can choose to either add or delete some links, which will change the topology of the whole network. If changes are made frequently enough, it will inevitably affect the network's performance. Therefore we are interested in whether there is an optimal strategy for each agent, collectively called the Nash equilibrium, a state such that each agent will not benefit from altering its current connection to the Internet.

In our investigation, we intend to expand on the work of [1] on certain areas, particularly on the Tree Conjecture, which in words says that all Nash equilibria are trees if the cost of building a new link is costly enough. But before that, we need to revisit the model proposed in [1]. We shall quote it verbatim from their paper.

4.2 The Model

Consider the model such that there are n players: $\{0, 1, \ldots, n-1\}$, which we denote by the set $[n]$. The strategy space of player i is the set $S_i = 2^{[n]-\{i\}}$. Given a combination of strategies $s = (s_0, \ldots, s_{n-1}) \in S_0 \times \ldots \times S_{n-1}$, we consider the graph $G[s]$, the underlying undirected graph of $G_0[s] = ([n], \bigcup_{i=0}^{n-1}(\{i\} \times s_i))$. The cost incurred by player i under s is defined to be

$$c_i(s) = \alpha|s_i| + \sum_{j=0}^{n-1} d_{G[s]}(i, j),$$

where α is cost of constructing an edge, s_i is the number of edges that are constructed by player i, and $d_{G[s]}(i, j)$ is the distance between nodes i and j in the graph $G[s]$. The social cost is defined to be the total of all players' costs. That is,

$$C(G) = \sum_i c_i(s) = \alpha|E| + \sum_{i,j} d_G(i, j).$$

A (pure) Nash equilibrium in this game is an s such that, for each player i, and for all s' that differ from s only in the i-th component, $c_i(s) \geq c_i(s')$. There can be more than one Nash equilibria for each specific value of α. The Price of Anarchy is defined to be the ratio

$$\text{Price of anarchy} = \frac{C(N)}{C(\text{OPT})}$$

where N is the worst case Nash equilibrium and OPT is the optimal graph.

4.3 Basic Results

We can achieve a trivial lower bound for the social cost based on the fact that every pair of nodes not connected by an edge is at least of distance 2 apart.

$$\begin{aligned} C(G) &\geq \alpha|E| + 2|E| + 2(n(n-1) - 2|E|) \\ &= 2n(n-1) + (\alpha - 2)|E|. \end{aligned}$$

This bound is valid for any graph with diameter at most 2.

Now we can examine various cases for the value of α and obtain some results on the social optimum and the worst case Nash equilibrium. These results will then be used to calculate the price of anarchy for the different values of α.

Case 1: $\alpha < 1$

Proposition 1. *The complete graph K_n is the only Nash equilibrium.*

Proof. Since $\alpha < 1$ the graph cannot have a diameter greater than 1. Otherwise, it would be more advantageous for a node u to create an edge between itself and a node v where $d(u, v) \geq 2$ (since the cost of adding an edge is less than the distance saved). However the only graph with diameter 1 is the complete graph K_n. Therefore the only possible Nash equilibrium is K_n. □

Corollary 2. *K_n is the optimal graph for $\alpha < 1$.*

Case 2: $\alpha = 1$

Notice that for this case any equilibrium will have a diameter less than or equal to 2. Otherwise, it would be beneficial for a node u to create an edge between itself and a node v where $d(u, v) \geq 2$.

Proposition 3. *The star graph is the worst case Nash equilibrium.*

Proof. The star graph is a tree with diameter 2. Call two of the terminal nodes u and v. Now the cost of adding the edge uv is $\alpha = 1$, but the savings in distance (for an individual node) is also 1. So any terminal node is indifferent towards creating an edge between itself and another terminal node and hence no such edges will be created. Therefore, the star graph is a Nash equilibrium.

It remains to show that for any other graph the social cost is greater than that of the star graph. Suppose that the star graph is not the worst case Nash equilibrium. Thus the worst case equilibrium is any graph, G, other than the star graph. The diameter of G will be either 1 or 2, as explained above, and G will contain at least one cycle. The only possible cycle then is the 3-cycle. Say that such a 3-cycle is made up of the nodes u, v, and w. Without loss of generality suppose that the edge uv is removed, and look at the effect on the social cost. The social cost of $C(G)$ will decrease by the amount $\alpha = 1$ for deleting the edge uv, but it will increase by 1 from $d(u, v)$ and by 1 from $d(v, u)$. So overall $C(G)$ will decrease by $\alpha = 1$ and will increase by 2. Hence a graph with fewer edges and a larger diameter will have a greater social cost than G. Therefore, for any graph G other than the star graph, $C(G) < C(\text{star})$, and the star graph is the worst case Nash equilibrium. □

Proposition 4. *The complete graph K_n is the optimal graph.*

Proof. Suppose not. Then some graph G, with n nodes, is the optimal graph. This implies that diam$(G) = 2$, as the only graph with diameter 1 is K_n. In the graph G there are two nodes, say u and v, where $d(u, v) = 2$. Create the edge uv and look at the effect on the social cost. The social cost $C(G)$ will increase by $\alpha = 1$ for the cost of adding the edge uv, but there will be a savings of 1 from $d(u, v)$ and a savings of 1 from $d(v, u)$. So overall $C(G)$ will increase by 1 and decrease by 2. Hence a graph with more edges and a lesser diameter than G will have a lesser social cost. Since G was an arbitrary choice (other than K_n), this implies that the complete graph K_n is the optimal graph. □

Theorem 5. *The price of anarchy is $\approx 4/3$.*

Proof. Here N is the star graph and OPT is the complete graph K_n. So

$$\begin{aligned}
\text{POA} &= \frac{\alpha(n-1) + (n-1)(2 + 2(n-2))}{\alpha(\frac{n(n-1)}{2}) + n(n-1)} \\
&= \frac{\alpha + 2n - 2}{\alpha\frac{n}{2} + n} \\
&\approx 2n/(3n/2) \\
&= 4/3.
\end{aligned}$$

□

Case 3: $1 < \alpha < 2$

Lemma 6. *The worst case Nash equilibrium does not contain cycles.*

Proof. Suppose not, then the worst case Nash equilibrium G contains at least one cycle. For $1 < \alpha < 2$ the diameter of any graph has to be less than or equal to 2, hence the only possible cycle is the 3-cycle. Call the nodes in one of the 3-cycles in G u, v, and w. Without loss of generality suppose that u is the buyer of the edge uw. Then in order to make it worthwhile for u to buy the edge there will be a node adjacent to w other than u and v, call this node x. But then the original graph contains the path $xwvu$, and this is the shortest path from x to u. Hence $d(x, u) = 3$ and the diameter of the original graph is 3, a contradiction. Therefore the worst case Nash equilibrium does not contain cycles. □

Proposition 7. *The star graph is the worst case Nash equilibrium.*

Proof. By the above lemma, the worst case Nash equilibrium does not contain any cycles. Hence the worst case Nash equilibrium must be a tree. But the only tree with diameter less than or equal to 2 is the star graph. Therefore the star graph is the worst case Nash equilibrium. □

For this case the complete graph K_n is no longer a Nash equilibrium, however it can be shown that it is still the optimal graph. Thus, as in the previous case, the price of anarchy is $\approx 4/3$.

Case 4: $\alpha \geq 2$

Lemma 8. *The contribution to $\sum_{i,j} d_G(i,j)$ by nodes with distance 1 is $2|E|$.*

Lemma 9. *The star graph is the optimal tree.*

Proof. It can be shown that

$$C(\text{star}) = \alpha(n-1) + 1 \cdot 2(n-1) + 2[n(n-1) - 2(n-1)]$$

Now

$$\begin{aligned} C(\text{tree other than star}) &\geq \alpha(n-1) + 1 \cdot 2(n-1) + 2[n(n-1) - 2(n-1) - k] + 3 \cdot k \\ &= \alpha(n-1) + 2(n-1) + 2[n(n-1) - 2(n-1)] + k \\ &> C(\text{star}), \end{aligned}$$

where k is the number of edges at a depth greater than 2. Therefore the star graph is the optimum tree. □

Theorem 10. *The star graph is the optimum graph.*

Proof. The social cost of a general graph is

$$\begin{aligned} &\geq \alpha(n-1+j) + 1 \cdot 2(n-1+j) + 2[n(n-1) - 2(n-1+j)] \\ &\geq \alpha(n-1) + 2(n-1) + 2[n(n-1) - 2(n-1)] - 2j + \alpha \cdot j, \end{aligned}$$

where j is the number of edges not found in the star graph ($j \geq 1$). Thus the difference in cost between the general graph and the star graph is $\alpha \cdot j - 2j = j(\alpha - 2)$. Therefore when $\alpha = 2$ the star graph is an optimal graph, and when $\alpha > 2$ the star graph is the optimal graph. □

It remains to be shown what the price of anarchy is for values of α greater than or equal to 2. However for this calculation the worst case Nash equilibrium must be known, and this is much more difficult to find than for the smaller values of α. The next few sections will provide a greater insight into this problem.

4.4 An Upper Bound on the Price of Anarchy

In [1], the author proved the following result for $\alpha \geq 2$.

Theorem 11. *For any tree Nash equilibrium T, $\rho(T) = C(T)/C(\text{star}) < 5$.*

Proof. See [1], p. 4. □

We would like to extend their result a bit further here. Fix $q_0 \in T$. We can partition the nodes by their distance from q_0. Let q_ℓ be a point that is of distance ℓ from q_0. Let d be the maximal distance ℓ can achieve, so $\ell \leq d$. Observe that, as mentioned in [1], $d < 2\sqrt{\alpha}$ since otherwise it would be more cost effective to build an edge between q_0 and the other node.

For any point $q \in G$, let

$$\varphi(q) = \sum_{q \neq q'} d_G(q, q')$$

which is the *cost of convenience* q contributes to the total cost. If $[q_\ell, q_{\ell+1}]$ is an edge, notice that since T is a tree,

$$\begin{aligned} \varphi(q_{\ell+1}) &= \varphi(q_\ell) + \text{No. of points closer to } q_\ell - \text{No. of points closer to } q_{\ell+1} \\ &< \varphi(q_\ell) + |E| \end{aligned}$$

and by induction we get $\varphi(q_\ell) < \varphi(q_0) + \ell|E|$. Let n be the total number of points in G, so $n = |E| + 1$. Let

$$p(\ell) = \frac{\text{No. of points at level } \ell}{n}.$$

Then the total cost $C(T)$ is given by

$$\begin{aligned} C(T) &= \alpha|E| + \sum_{\ell=0}^{d} np(\ell) \cdot \varphi(q_\ell) \\ &= \alpha|E| + \varphi(q_0) + \sum_{\ell=1}^{d} np(\ell) \cdot \varphi(q_\ell) \\ &< \alpha|E| + \varphi(q_0) + \sum_{\ell=1}^{d} np(\ell)(\varphi(q_0) + \ell|E|) \\ &= \alpha|E| + \varphi(q_0) + n\varphi(q_0)\sum_{\ell=1}^{d} p(\ell) + n|E|\sum_{\ell=1}^{d} \ell p(\ell) \\ &= \alpha|E| + \varphi(q_0) + n\varphi(q_0)\frac{n-1}{n} + n(n-1)E[\ell] \\ &= \alpha(n-1) + n\varphi(q_0) + n(n-1)E[\ell] \\ &< \alpha(n-1) + 4\sqrt{\alpha}n(n-1) \end{aligned}$$

where $E[\ell] \leq d$ is the average value of ℓ and note that $\varphi(q_0) < d(n-1)$.

For $\alpha \geq 2$, star graph is the optimal graph, and $C(\text{star}) = \alpha(n-1) + 2(n-1) + 2(n-1)(n-2)$. Therefore

$$\rho(T) = \frac{C(T)}{C(\text{star})} < \frac{\alpha n + 4\sqrt{\alpha}n(n-1)}{\alpha(n-1) + 2(n-1) + 2(n-1)(n-2)} = F(T).$$

For $\rho(T) < K$, it suffices to set $F(T) < K$ and it gives

$$(2K - 4\sqrt{\alpha})n^2 + O(n) > 0.$$

This is a quadratic equation with leading coefficient $2K - 4\sqrt{\alpha}$. So for n sufficiently large, its sign depends only on the sign of $2K - 4\sqrt{\alpha}$. Therefore we have the following result

Proposition 12. *For $2 \leq \alpha < K^2/4$ and n sufficiently large, $\rho(T) < K$.*

For instance, if we set $K = 4$, we have that $\rho(T) < 4$ for $2 \leq \alpha < 4$ and n sufficiently large. In the next section we shall examine some numerical results we have on the price of anarchy.

4.5 Numerical Results

In [1], the authors proved the following result

Theorem 13. *For any $\epsilon > 0$, there exists a Nash equilibrium of the network game with the price of anarchy greater than $3 - \epsilon$.*

For any $k \geq 4$ (the degree of a graph) and $d \geq 2$ (the depth of a graph), the family of complete directed trees with all edges going from parent nodes to their children provided the key to the theorem above. It has been shown in [1] that this graph is a Nash equilibrium if α is large enough. For instance, we consider $\alpha = (d-1)n$, where n is the number of nodes. We have deduced a general formula for computing the total cost for any k and d.

Proposition 14. *The cost of a complete directed tree with all edges going from parent nodes to their children is given by the following formula*

$$\alpha n + \sum_{j=1}^{d} \sum_{m=1}^{d-j+1} k^{d-j+1} \left[k^{d-m+1}(d-j+2-m) + (k-1)S(d-m+1) \right] + k^{d-j+2} S(j-1)$$

where $S(r) = (k^r(r+2) - 2)/(k-1)$.

Proof. Imagine drawing our graph in a downward fashion so that all the points of generation d are lining up on the d-th line, linking up to their parents on the $d-1$-th line. Let us consider the very left point of generation d. We may split the graph into two parts. One part consists of starting nodes. Excluding the very left node of the first generation, there are $k-1$ nodes of the first generation, and all nodes going from these $k-1$ nodes. Remaining part will be another tree with depth $d-1$. The sum of distances from any point of generation d to all the nodes from the first part can be computed by

$$d + (d+1)(k-1) + (d+2)(k-1)k + \ldots + (d+d)(k-1)k^{d-1}.$$

Now we split remaining part, which is also a ”regular” tree, into two parts again and repeat the same computation. Since we have k^d points of generation d we get that the total cost of all points of generation d is given by

$$\sum_{n=0}^{d-1} k^d \left[(d-n) + (d-n+1)(k-1) + \ldots + (d-n+d-n)(k-1)k^{d-n-1} \right].$$

Similarly for the points of generation $d-1$ we get for each point

$$(d-1-n) + (d-1-n+1)(k-1) + \ldots + (d-1-n+d-n)(k-1)k^{d-n-1}.$$

And the cost of all points of generation $d-1$ is

$$\sum_{n=0}^{d-2} k^{d-1} \left[(d-1-n) + (d-1-n+1)(k-1) + \ldots + (d-1-n+d-n)(k-1)k^{d-n-1} \right] + k^{d-1}k.$$

In general for each point of generation $d-i$ we get

$$(d-i-n) + (d-i-n+1)(k-1) + \ldots + (d-i-n+d-n)(k-1)k^{d-n-1}.$$

The cost of all points of generation $d-i$ is

$$\sum_{n=0}^{d-i-1} k^{d-i} \left[(d-i-n) + \ldots + (d-i-n+d-n)(k-1)k^{d-n-1} \right] + k^{d-i} \sum_{j=1}^{i} jk^{j-1}.$$

At last we just have to add everything up. Using simple geometric series formula, and noting that the series $\sum_{j=1}^{i} jk^{j-1}$ is a derivative of the geometric series we get the desired formula. □

Here are some numerical results. They suggest that to get the worst scenario we should increase the depth. If we fix the depth and increase the degree, we construct ”star-type” network. Therefore, the price of anarchy will be increasing rather slowly.

$d(depth)$	$k(degree)$	POA	$d(depth)$	$k(degree)$	POA
100	4	2.9703	4	100	2.5939
200	4	2.9851	4	200	2.5970
250	4	2.9880	4	500	2.5976

Table 4.1: Comparison of the anarchy price for regular trees

4.6 The Tree Conjecture

It is stated in [1] that the Peterson graph is a transient equilibrium for $\alpha \leq 4$. A transient Nash equilibrium is an equilibrium where there exists at least one node that could add or subtract an edge at no cost to that node. The authors also proposed the following conjecture:

Conjecture 15 (The Tree Conjecture). *There exists a constant A, such that for $\alpha > A$, all non-transient Nash equilibria are trees.*

Although we have not proved the conjecture in general, we did find two interesting results.

Proposition 16. *3-cycles cannot exist without the presence of other cycles in a non-transient Nash equilibrium. (See Figure 4.1)*

Proof. First some terminology: Let the nodes on a three cycle be labelled a_1, a_2, a_3. Now there exist 4 sets of nodes that could have an edge adjacent to more than one node on the cycle $a_1a_2a_3$. Let these sets be call T_{12}, T_{13}, T_{23}, and T_{123}. The subscripts represent which nodes that the set attaches to. Attached to each node there may exist a subgraph that is rooted by a node to the cycle. These subgraphs will be denoted as A_1, A_2, and A_3 respective to a_1, a_2, and a_3.

Assume that such a graph exists. Without loss of generality, we assume a_1 buys the edge connected to a_2. This implies that $|A_2| > \alpha$ since it is a non-transient equilibrium. This implies that $|A_1| = |A_3| = 0$ since any node in either A_1 or A_3 would connect itself to a_2 and then be in one of T_{12}, T_{23}, or T_{123}.

Again without loss of generality we can assume a_1 bought the edge from a_1 to a_3. If a_1 removes this edge, then the cost function of a_1 decreases by $\alpha - 1$. There a contradiction that this was indeed a Nash equilibrium for $\alpha > 2$. □

We also proved the tree conjecture in the case that every cycle has a common edge with another cycle. The techniques involved here can hopefully be extended to more general cases.

Proposition 17. *A cycle cannot exist without having a common edge with another cycle in a non-transient Nash equilibrium. (See Figure 4.2)*

Proof. As in the proof above A_i will stand for the respective subgraph rooted to the cycle at a_i and as before a_i is contained in A_i. Assume that there is such a graph, and we shall reach for a contradiction.

There are 2 possible cases: (i) Every node on the cycle is the buyer of one edge on the cycle, and (ii) There exists a node that bought both of the edges adjacent to it on the cycle.

Case (i). For this to happen it is easy to see that, without lose of generality, the edges bought for a k-cycle can be written as a_i buys a_ia_{i+1} and a_k buys a_ka_1. Now there exists a maximum $|A_i|$. Also without loss of generality we can set A_3 as the maximum. Now $|A_2| < |A_3|$ and a_1 has bought edge a_1a_2. But the cost of a_1 goes down by $|A_3| - |A_2|$ if a_1 has a_1a_3 in its edge set instead of a_1a_2. Thus a contradiction.

Case (ii). Without loss of generality we can assume a_1 is the double buyer. a_1 has bought both edge a_1a_k and a_1a_2. If a_1 has bought edge a_1a_2 this implies $|A_2| > 2|A_3|$ else a_1 would buy edge a_1a_3 and not have bought a_1a_2. Since $|A_2| > 2|A_3|$ a_3 buys edge a_3a_4 else a_4 would buy edge a_2a_4 instead of a_3a_4. The same argument can be extended to show that a_j buys a_ja_{j+1} for all j odd, $j \leq k$. It is clear to see that if k is odd there exists a contradiction in that a_1 and a_k would both have bought edge a_1a_k. So now it remains to be shown the contradiction on the even cycle.

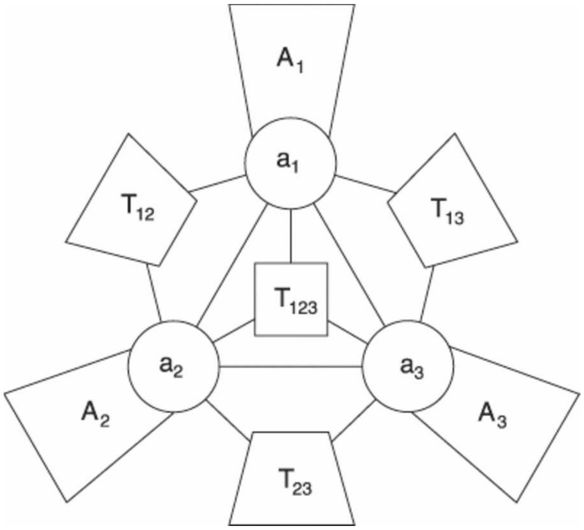

Figure 4.1: For Proposition 16.

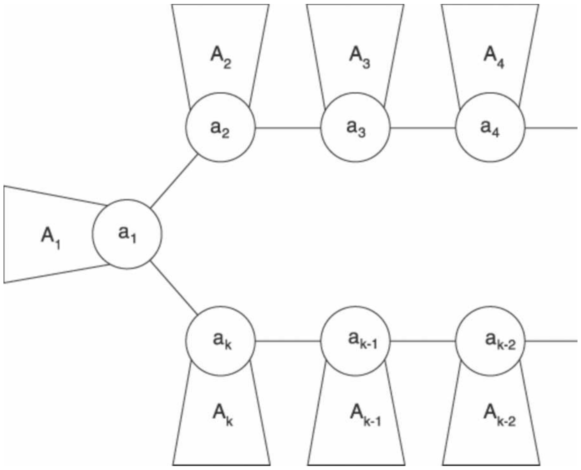

Figure 4.2: For Proposition 17.

First it is instructive to also examine the effect of a_1 buying the edge a_1a_k on the cycle. Similar to the previously stated argument, $|A_k| > 2|A_{k-1}|$, it also forces all a_j to buy edge a_ja_{j-1} for all j odd. What is created is an alternating sequence of double buyers on the cycle.

Now examine $|A_2|$ and $|A_4|$. Without loss of generality we can set $|A_2| \leq |A_4|$. If $|A_2| < |A_4|$, a_1 will buy edge a_1a_4 instead of a_1a_2 since this would reduce a_1's cost function. If $|A_2| = |A_4|$ the graph is at best transitive. This completes the proof. □

4.7 Ending Remarks

In the past week we have closely examined the model presented in [1], came up with most of the results in [1] independently, and in some cases expanded upon their results.

For future work, we recommend continuing on proofing the Tree Conjecture. It is also interesting if we can propose a slightly different cost function, as the current one doesn't capture the possibility of co-contribution for building each new connection, neither does it take in spatial distribution of vertices into account.

Bibliography

[1] A. Fabrikant, A. Luthra, and E. Maneva, On a Network Creation Game. Annual ACM Symposium on Principles of Distributed Computing archive, (2003) 347–351.

[2] E. Koutsoupias and C.H. Papadimitriou. Worst-case equilibria. Lecture Notes in Computer Science, 1563: 404–413, 1999.

Chapter 5

Path Planning for an Autonomous Robot

Participants: Randall Pyke (Mentor), Nancy Azer, Karel Casteels, Zhu Jiaping, Yury Petrachenko, Mahin Salmani, Tzvetalin Vassilev, Mengzhe Wang, Pengpeng Wang, Mohammed Ali Yassaei, Bo Zeng.

PROBLEM STATEMENT: The problem considered here is to plan a path for an autonomous robot through a lattice, travelling from node to node along the arcs of the lattice while visiting a prescribed subset of the nodes (an itinerary). A model of such a situation is a robot performing a security run inside a building. The robot has information about the lattice (eg., locations of nodes and arcs), including information which may be updated during the trip (eg., an arc becomes impassable). For variants of this basic problem, we describe algorithms that tell the robot which path to follow through the lattice to carry out its itinerary in the least amount of time.

5.1 Introduction

Consider an autonomous security robot who must visit certain rooms in a building in the most efficient manner possible (i.e., in the least amount of time). We model the building as a graph where each node represents a room in the building and the edges represent the hallways. The edges are weighted by the distance between rooms, or equivalently, by the time needed for the robot to travel between rooms (other ‘cost functions’ could be used).

We will consider several scenarios. In Section 5.2 the situation is that of a single robot. We are given an itinerary of nodes and we wish to find a least cost (least distance) path between the robot’s starting node and its desired end node such that each node in the itinerary is visited at least once.

Such a problem can be easily modelled as a linear programming problem and solved using standard methods which are described in the following sections.

In Section 5.3 we discuss the case of finding a least cost path using multiple robots. Here the robots’ positions are fixed before the itinerary is given. We give a heuristic method to solve this problem and a method that uses integer programming (IP).

Section 5.4 addresses the same situation as in Section 5.3 except that now the initial positions of the robots are to be determined (the number of robots available is fixed a priori). First we consider the case when there is no a priori information on the possible itineraries, and thus distribute the robots initially so that they minimize the total distance between the robots and the nodes of the network. We then mention how this can be modified to accommodate any a priori information on the itineraries by weighting the nodes according to how likely they are to be included in an itinerary.

In Section 5.5 we further generalize the situations discussed in Sections 5.3 and 5.4. Now we also decide on the number of robots that are to be used to carry out the itineraries.

Section 5.6 returns to the case of a single robot. In this realistic scenario each arc has a probability that it is unusable. The problem now is: how do we determine an optimal path in the sense of it being short and likely to be passable?

Finally, we mention that the algorithms proposed here are not specific to rectangular lattices but can be implemented on networks with more general underlying graphs. The rectangular lattices presented here are relevant to the particular problem we are considering, that of a robot performing a security run in a building or city.

5.2 Single Robot Optimal Path

5.2.1 Introduction

Here we have a single robot starting from a node r and we wish to find the least cost path to the end node s, with the provision that we also visit each node in a given itinerary $\mathcal{I} = \{i_1, i_2, \ldots, i_M\}$. The order in which the robot visits the itinerary nodes is left for us to decide.

There are two steps to the algorithm. First we find the least cost path between every pair of nodes in the itinerary. In the second step, we solve a travelling salesman problem on a complete graph which we call the *shortest path graph* $\tilde{G}$. The vertices of $\tilde{G}$ are the itinerary nodes and the edges between these nodes are weighted by the length of the shortest path between them in the original network.

This problem was first posed by Martin Talbot of Ryerson University[1], whose methodology we follow here [1].

5.2.2 LP Formulation of Trivial Shortest Paths

Let c_{ij} be the cost of the edge between node i and j. For any path, let us introduce the variable x_{ij}, where $x_{ij} = 1$ if the edge ij (the edge joining nodes i and j) is used in the path, and 0 otherwise. The problem of finding the shortest path between two nodes r and s can be formulated as the following linear program.

[1] Now currently at the University of Waterloo; email: `talbotm@acm.org`.

$$\text{minimize} \qquad \sum_{i,j} c_{ij}x_{ij}, \tag{5.1}$$

subject to:

$$\begin{aligned}
\sum_i (x_{ri} - x_{ir}) &= 1 && (5.2)\\
\sum_i (x_{is} - x_{si}) &= 1 && (5.3)\\
\sum_i x_{ij} - \sum_k x_{jk} &= 0, \qquad \forall j \neq r, s. && (5.4)
\end{aligned}$$

Constraint (5.2) says that we must leave the start node r and similarly, (5.3) implies that we must finish at the end node s. Conversely, (5.4) says that if we are at a node which is neither r nor s, we must leave it again. The sums are over all the nodes in the network.

5.2.3 Concatenation of Shortest Paths and the TSP

Using the trivial shortest path algorithm established in the previous section we compute the shortest path between all pairs of nodes in the itinerary $\widetilde{\mathcal{I}} = \{r, s\} \cup \mathcal{I}$. We then construct the complete graph on $|\widetilde{\mathcal{I}}| = n$ vertices and weight the edge between nodes i and j by the cost of the shortest path between them, which we denote as $\widetilde{c}_{ij}$. We call this network the *shortest path graph, $\widetilde{G}$.*

The final step amounts to a travelling salesman problem (TSP) on $\widetilde{G}$ (for a description of TSP, see for example [2]). That is, we wish to find a least cost Hamiltonian cycle for $\widetilde{G}$. We solve this using another linear program, with the proviso that it is effective only for approximately $n \leq 100$ itinerary nodes since the TSP LP algorithm is exponentially complex. For larger problems we may use one of a number of heuristic methods, achieving a good, but not necessarily optimal, solution, with the advantage of running the algorithm in polynomial time.

For any path, let us use the variable $\widetilde{x}_{ij}$ where $\widetilde{x}_{ij} = 1$ if the edge ij is included in the path and 0 otherwise. Thus we want to

$$\text{minimize} \qquad \sum_{i} \widetilde{c}_{ij}\widetilde{x}_{ij}, \tag{5.5}$$

subject to:

$$\begin{aligned}
\sum_i \widetilde{x}_{ij} &= 1, \qquad \forall j && (5.6)\\
\sum_i \widetilde{x}_{ji} &= 1, \qquad \forall j && (5.7)\\
\sum_i \widetilde{x}_{ij} - \sum_k \widetilde{x}_{jk} &= 0, \qquad \forall j \neq r, s. && (5.8)
\end{aligned}$$

The only remaining difficulty is that the solutions to the above linear programs may contain a disjoint cycle. We work around this problem by adding another constraint, due to Miller, Tucker and Zemlin [2, 3]. Let the variable u_i be a number between 1 and N, where N is the number of nodes in the graph and such that $u_1 = 1$ and $2 \leq u_i \leq N$ for $i \neq 1$. Then to the above LP problem we add the constraint

$$u_i - u_j + 1 \leq (N-1)(1 - \widetilde{x}_{ij}), \qquad \forall i \neq r, \forall j \neq s. \tag{5.9}$$

The constraint (5.9) works in the following way. First note that if for some edge ij we have $\widetilde{x}_{ij} = 1$, then $u_j \geq u_i + 1$. Now suppose our solution contained a cycle disjoint from the rest of the tour. Without loss of generality, we assume this cycle does not contain the start node r. Then if we calculate the values of u_i for the edges in this cycle, we get an an increasing sequence, such that eventually some $u_i \rangle N$, contradicting our bound that $u_i \leq N$. Thus this "solution" violates the constraint.

One may make the objection that in the set of shortest paths we might, say, along the shortest path ij between the two itinerary nodes i and j, visit the itinerary node k so that we need not visit k again in the second part of the algorithm when we concatenate the shortest paths to visit all nodes. Fortunately, this argument can be resolved in the following way. If this happened, then it must be that

$$\widetilde{c}_{ij} = \widetilde{c}_{ik} + \widetilde{c}_{kj}.$$

Now consider the complete graph constructed in the second part of the algorithm. Suppose we have a solution to the travelling salesman problem which includes the edge ij. Then, since we still need to visit k at some point, we have the path P_{jk} and the path P_{ks}. Now consider a solution to the travelling salesman problem which includes the edges ik and kj. From node j, we have the path Q_{js} which finishes off the solution. Suppose

$$\widetilde{c}_{ij} + \sum_{lm \in P_{jk}} \widetilde{c}_{lm} + \sum_{lm \in P_{ks}} \widetilde{c}_{lm} \leq \widetilde{c}_{ik} + \widetilde{c}_{kj} + \sum_{lm \in Q_{js}} \widetilde{c}_{lm}.$$

Then we must have

$$\sum_{lm \in P_{jk}} \widetilde{c}_{lm} + \sum_{lm \in P_{ks}} \widetilde{c}_{lm} \leq \sum_{lm \in Q_{js}} \widetilde{c}_{lm},$$

so that the path $P_{jk} + P_{ks}$ has an equal or shorter length than path Q_{js}. At worst, this contradicts the choice of Q_{ks} and at best we conclude it doesn't make a difference whether we use the edge ij and then revisit k or use the edges ik and kj.

5.2.4 Heuristic Method For Solving TSP

For large values of $n > 100$, we may need to use a heuristic method to solve the TSP since in general, no known algorithm exists to solve the TSP with worst case running time less than O(2^n). An example of a good heuristic method is the Nearest Neighbour algorithm. Here we start at any node and choose as the next node in our path the nearest node not yet visited. Since our graph satisfies the triangle inequality

$$c_{ik} + c_{kj} \geq c_{ij} \quad \forall i, j, k \in \widetilde{\mathcal{I}},$$

it can be shown (see $[1]$) that this algorithm's worst case tour has length $\frac{1}{2}\lceil \log_2(n) \rceil + \frac{1}{2}$ times the length of the optimal shortest path, although tests have shown the average solution to be only 1.26 times the optimal.

5.2.5 Example of a Solution

Suppose we are given the network in Figure 5.1, where the start and end node is node 1 and the itinerary nodes are shaded in grey. Figure 5.2 shows the optimal path,

$$1 \to 5 \to 6 \to 7 \to 3 \to 4 \to 8 \to 12 \to 16 \to 15 \to 14 \to 13 \to 14 \to 10 \to 6 \to 2 \to 1,$$

giving us a least cost path of length 51.

Applying the nearest neighbour algorithm we get a path with cost 55.

5.3 Multiple Robots; Number of Robots Fixed, Initial Locations Fixed

5.3.1 Problem Definition (Problem A)

Imagine a situation where we have multiple robots to travel an itinerary. We want to plan a set of closed paths[2], or circles in short, each for every robot, such that the union of the paths will cover the itinerary and at the same time, the total distance, the sum of all the travelling distances of all the robots, is minimized.

[2] "Closed" here means that the paths start and end at the same node.

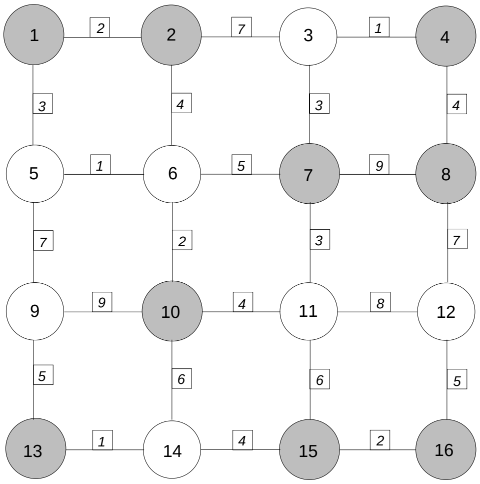

Figure 5.1: Example graph with itinerary nodes shaded.

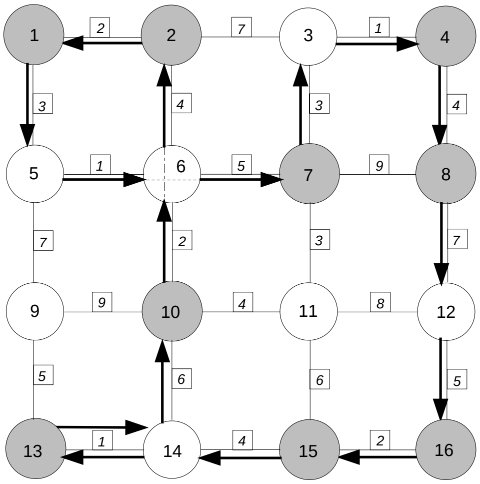

Figure 5.2: Solved example graph with itinerary nodes shaded.

In Problem A of this situation, we assume that both the number of robots and their initial start points/nodes, and hence their end points/nodes, in the graph are fixed.

Here we give the formal mathematical form of this problem.

A bi-directed graph G is defined as $G = (V, E)$ in which $V = \{1, \ldots, n\}$ is the set of all nodes in the graph and $E = \{e_{ij}, 1 \leq i \neq j \leq n\}$ is the set of all edges. Each $e_{ij} = \langle i, j\rangle$ represents an arc that the robot can travel from node i to j. There is also a cost function $cost(e_{ij}) = c_{ij}$ defined on each edge e_{ij} as the distance travelled by the robot along e_{ij}.

For a given graph G, we also define the set of m itinerary nodes, $\mathcal{I} = \{I_l, 1 \leq l \leq m\}$, as the set of the nodes that have to be travelled by at least one robot at least once. There is a set of robots, $R = \{r_k, 1 \leq k \leq K\}$, where robot r_k starts and ends at node p_1^k, are given to travel the itinerary. In Problem A the number and initial locations of the robots is fixed. In Problem B, the number of robots is fixed, but their initial locations is to be determined.

Given these settings, Problem A is to find a set of closed paths or circles, p_k, $1 \leq k \leq K$, where p_k is assigned to robot r_k, and where $p_k = (p_1^k, p_2^k, \ldots p_{end_k}^k)$ and $p_{end_k}^k = p_1^k$. (In the above, we assume that all the nodes in the path are in the graph, i.e., $p_1^k, p_2^k, \ldots p_{end_k}^k \in V$.) The goal is to find paths for each robot such that they cover the itinerary and that the total distance travelled by the robots is minimized;

$$\min_{p_k,\ 1 \leq k \leq K} \sum_k cost(p_1^k, p_2^k) + cost(p_2^k, p_3^k) + \cdots + cost(p_{end_k - 1}^k, p_1^k).$$

5.3.2 Solution

Transformation from Problem A to a TSP with r Robots

Based on the observation that if we take the number of robots to be 1 and all the nodes in the network to be the itinerary nodes, Problem A is reduced to the “TSP with multiple visits (TSPM)”. Thus Problem A is at least as hard as the TSPM. Inspired by this, our solution to Problem A is first to transform it to the TSPM with r robots.

The transformation is done by constructing a different graph $\tilde{G} = (\tilde{V}, \tilde{E})$ with a new cost function defined on $\tilde{E}$, in which $\tilde{V}$ is the union of the itinerary nodes and robot starting nodes, and the edge $\tilde{e}_{ij} = \langle \tilde{i}, \tilde{j}\rangle \in \tilde{E}$ exists if there exists a shortest path between $\tilde{i}$ and $\tilde{j}$ in the original graph G. The cost of this edge is the cost of this shortest path. We call the new graph $\tilde{G}$ the shortest path graph.

The algorithm of this transformation is as follows,

Algorithm: Transformation from problem A to TSP with r robots

```
$\tilde{V} = I$ % such that $\tilde{v}_i = I_i, 1 \leq i \leq m$
$\tilde{E} = \phi$
for $1 \leq i \neq j \leq m$
  % using the shortest path algorithm
  Compute the shortest path in $G$ between $I_i$ and $I_j$ with distance $length(I_i, I_j)$
  if $length(I_i, I_j)\langle\infty$
    $\tilde{E} = \tilde{E} \cup \tilde{e}_{ij}$
    $\tilde{c}_{ij} = cost(\langle\tilde{i}, \tilde{j}\rangle) = length(I_i, I_j)$
$\tilde{G} = (\tilde{V}, \tilde{E})$
```

The problem now is to find circles, one for each robot, that starts and ends at its specific starting node, and to travel all the nodes in $\tilde{V}$ exactly once. We call this problem the “Multi-robot TSP” problem. After solving this “Multi-robot TSP”, we can easily transform this solution, a sequence of itinerary nodes, back to the solution to the original problem. This is done by adding between two consecutive itinerary nodes in the solution on the shortest path graph $\tilde{G}$ all the edges in the original network G this path contains.

In the following, we are going to discuss how to solve this “Multi-robot TSP”.

Solution to the “Multi-robot TSP”

Again, we should re-emphasize here that the problem now is for all the robots to travel exactly once all the nodes in the shortest path graph, $\tilde{G}$.

We present three solutions; one based on heuristics, one based on linear programming (LP), and one based on the transformation of the problem to a TSP problem on a different graph.

1. Jiaping's Algorithm

The first solution was contributed by Jiaping. We call it "Jiaping's Algorithm". Jiaping's Algorithm is based on the heuristic of the one robot itinerary solution. Intuitively, if one robot can travel all the itinerary nodes by following the path p' in reasonable time, a partition of p' into K parts, one for each robot, would also be reasonably short.

Based on this, Jiaping's algorithm is as follows. First, solve the TSP on $\tilde{G}$. We denote the solution by p' [3]. Now partition p' into r pieces, each piece assigned to one robot. The partition is done by cutting the longest edge between two consecutive robot nodes in p'. The itinerary nodes connected to robot r_k after cutting are assigned to the set of nodes designated for r_k to travel. Then, for each element of the partition we solve the TSP using the method of Part 2 of our report.

This longest cost cutting heuristic may not give us the optimal solution as we can see in some simulations.

2. Solution based on Integer Linear Programming (IP)

Another possible solution is to formulate the "Multi-robot TSP" as an integer programming (IP) problem. In the IP formulation, we define a set of $0/1$ binary variables $\tilde{x}^k_{ij}$, where i, j labels the edges in $\tilde{G}$ and k labels the robots. $\tilde{x}^k_{ij}$ denotes whether the edge $\tilde{e}_{ij}$ is travelled by robot r_k ($\tilde{x}^k_{ij} = 1$) or not ($\tilde{x}^k_{ij} = 0$). The IP problem contains the following statements (5.10)–(5.14).

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} \tilde{x}^k_{ij},\ where\ \tilde{x}^k_{ij} \in \{0,\ 1\}. \tag{5.10}$$

$$y^k_i \in \{0,\ 1\} \tag{5.11}$$

(5.10) describes our aim to minimize total distance travelled by the robots. (5.11) indicates whether the k^{th} robot visits the i^{th} node. Thus, the variables y^k_i describe a partition of the nodes amongst the robots.

$$\sum_k y^k_i \geq 1 \text{ for each } i. \tag{5.12}$$

This guarantees that every node is travelled at least once by a robot.

$$\widetilde{x^k_{ij}} \leq y^k_i \text{ for each } j. \tag{5.13}$$

This guarantees that if the k^{th} robot visits the i^{th} node, indicated by $y^k_i = 1$, it travels along some adjacent edge to this node.

Then we add the subtour elimination constraints:

$$u^k_i - u^k_j + N \cdot \widetilde{x^k_{ij}} \leq N - 1, \text{for } i, j \neq \text{ robot starting or ending nodes.} \tag{5.14}$$

The u's in the above equation are variables.

3. Transformation from "Multi-robot TSP" to TSP

This approach is proposed by Pengpeng, based on the following observation. If we break the circles of the solution to Multi-robot TSP, and chain them together to make a single circle such that the end of the previous piece is connected to the start of the next piece by a zero cost edge, the total cost of the Multi-robot TSP solution will be the same as this single circle.

The algorithm is as follows;

[3] As we can easily see, the choice of r' to start with does not affect the result, i.e., we will have the same p' no matter which robot is used.

Algorithm: Transformation from Multi-robot TSP to TSP

```
$\tilde{V}' = \tilde{V}$
$\tilde{E}' = \tilde{E}$
% Remember the robots' starting nodes are called $\tilde{v}_1^k$, $1 \leq k \leq K$
Make a copy of $G' \cap R$
Name the copy $RV = rv_k, 1 \leq k \leq K$ % the virtual node set
% Add virtual nodes to the original graph
$\tilde{V}' = \tilde{V} \cup RV$
% Connect virtual nodes to their corresponding $\tilde{v}_1^k$ nodes by 0-cost edge
Make edges $\langle \tilde{v}_1^k, rv_k \rangle$, $1 \leq k \leq K$
$\tilde{E}' = \tilde{E}' \cup (\cup_k \langle \tilde{v}_1^k, rv_k \rangle)$
$cost(\langle \tilde{v}_1^k, rv_k \rangle) = 0, 1 \leq k \leq K$
% Connect virtual nodes to the nodes that are connected to their corresponding nodes in $\tilde{G}$
if $\langle \tilde{v}_1^k, \tilde{v}_{i'} \rangle \in \tilde{E}$
    Make edge $\langle rv_1^k, \tilde{v}_{i'} \rangle$
    $\tilde{E}' = \tilde{E}' \cup \langle rv_1^k, \tilde{v}_{i'} \rangle$
    $cost(\langle rv_1^k, \tilde{v}_{i'} \rangle) = 0$
% Make the complete graph of robot starting nodes with zero-cost edges
for $1 \leq k \neq k' \leq K$
    Make edge $\langle \tilde{v}_1^k, \tilde{v}_1^{k'} \rangle$
    $\tilde{E}' = \tilde{E}' \cup \langle \tilde{v}_1^k, \tilde{v}_1^{k'} \rangle$
    $cost(\langle \tilde{v}_1^k, \tilde{v}_1^{k'} \rangle) = 0$
% Make the complete graph of robot starting nodes with zero-cost edges
for $1 \leq k \neq k' \leq K$
    Make edge $\langle rv_1^k, rv_1^{k'} \rangle$
    $\tilde{E}' = \tilde{E}' \cup \langle rv_1^k, rv_1^{k'} \rangle$
    $cost(\langle rv_1^k, rv_1^{k'} \rangle) = 0$
$\tilde{G}' = (\tilde{V}', \tilde{E}')$
```

Then we need to solve the TSP on $\tilde{G}'$ and transform the solution p' back to the initial problem. The algorithm is as follows. First, we collapse the virtual nodes and their corresponding robot starting nodes together. Then there will be one circle at each robot starting node (Remember this is a TSP so the sum of the numbers of visits at each robot starting node and its virtual node is exactly two.), if they are not connected by a 0-cost edge. If a robot node and its virtual node are actually connected by a 0-cost edge, our algorithm will make this robot stay there and not travel.

So we only consider how to provide paths for those travelling robots. First, these robots are scheduled to travel its circle. Then, for two adjacent robot starting nodes on the collapsed version of p', we can pick either of the robots to travel the piece on collapsed p' between them and then go back by the shortest path to its starting nodes.

After discussions, we realized that the paths we got from this are not necessarily the optimal solution to Multi-robot TSP. However, in some cases where there is no other node between two adjacent robot starting nodes on the collapsed p', the solution given above is optimal. The reason is as follows. For the optimal solution of Multi-robot TSP, we can make virtual nodes and chain them by 0-cost edges as mentioned before. The circle made then is a solution to the Multi-robot TSP. And at the same time, the transformation mentioned above can make the optimal solution to the Multi-robot TSP to a solution to TSP. So it must be optimal to TSP too.

But for cases where there are other nodes between two adjacent robot starting nodes on the collapsed p', the transformed solution we get is not optimal. However, we can say something on the solution. That is, the solution is bounded by the cost of the optimal solution plus the sum of the shortest distances between robot starting nodes. This can be intuitively explained by the fact that the sum of the cost of the circles is less than or equal to the cost of the optimal solution. And for the robot to be able to travel back to its starting node, the additional cost is at most the sum of the shortest distances between robot starting nodes.

Heuristic Solution to Problem A

TSP is an NP-complete problem and is well known that exponential computational time is needed to obtain an optimal solution (unless “NP $=$ P”). This is why for $n > 100$ we would compromise for a sub-optimal solution for the sake of computational time. Here we are going to talk about how to solve the problem by heuristics. The issue here is to get a ‘good’ solution without spending a lot of computational time. We realize that the solution obtained in this way may not be optimal, but we try to estimate how close to optimality it is.

The strategy we use is called the “Heuristic Partition”. Again, first we compute the shortest path graph $\tilde{G}$. Then assign each itinerary node to its nearest robot. After the assignments of all the nodes, we obtain a partition of the itinerary, one element of the partition for each robot. We can then solve the multiple robot problem by solving the single robot problem on each element of the partition as discussed in Part I of our report.

Formally, the algorithm is as follows. (In the following, we denote the robot starting nodes as $\tilde{v}_1^k$, to be consistent with our former nomenclature.)

```
Algorithm: Heuristic Partition
% The partition sets
$Part_k = \phi, 1 \leq k \leq K$
Compute the shortest path graph $G = (V, E)$ with costs $c_{i,j}$
For each $\tilde{v}_i \in \tilde{V}$
    % Choose the robot $r_{k'}$ such that $cost(\langle v_1^{k'}, v_i \rangle)$ is minimal
    $r_{k'} = \arg\min_{r_{k'}} cost(\langle v_1^{k'}, v_i \rangle)$
    $Part_{k'} = Part_{k'} \cup r_{k'}$
% Solve the one-robot planning problem for each partition
for $k = 1 \ldots K$
    Solve one-robot planning problem for $Part_k$
```

In solving the one-robot planning problem for each partition, we can also use some heuristics as mentioned in Part I of our report in the case of a large number of nodes.

If we use the exact solver in Part I to solve the one-robot planning problem, we know that our solution is better than using a single robot to travel all the itinerary nodes. However, the overall performance will depend on the quality of the partition and we cannot say anything on this. One more comment we would like to mention here is that this heuristic can be used to partition not only the itinerary nodes but the entire collection of nodes of the original graph. This will be especially useful for cases where we do not know what the itinerary nodes are a priori. A particular application of this would be for example in exploration, where we obtain the map in advance but the tasks of where to explore are given differently from time to time. This heuristic partition can then give us the same partition for different tasks and save a lot of computational time.

5.3.3 Simulation Results

To test the efficiency of the algorithms, we implemented them in Matlab on the problems described in Figures 5.4 and 5.5.

Also, the single robot result is shown in Figure 5.3 for comparison with the multiple robot planning cases.

The two problems above are based on the same graph with the same costs. However, the second one is for three robots starting at nodes 1, 4 and 13. (The itinerary nodes are coloured grey.)

It is interesting to see that both of the heuristic partition based algorithms gave the same result as in Figure 5.4.

The IP formulation implemented by the Matlab optimization toolbox gives the optimal solution as in Figure 5.5. It is interesting to see that the optimal solution in this case has the robot at node 13 remaining fixed, while the robots starting at nodes 1 and 4 travel as in the figure.

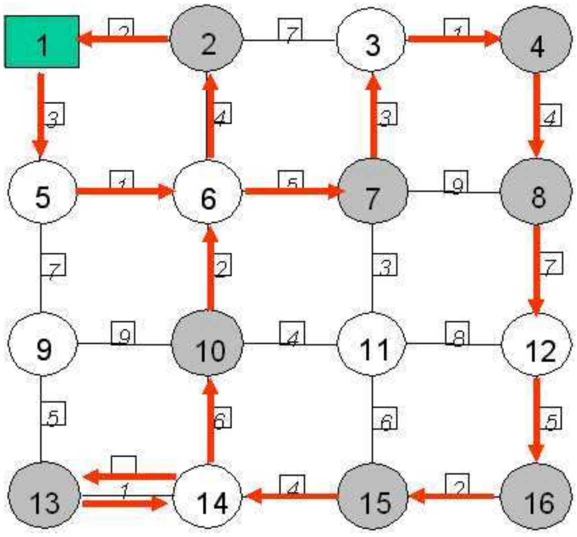

Figure 5.3: Single Robot planning result using algorithm in Part I. The robot starts at node 1, denoted by square, travels as directed by the arrows, and ends at node 1.

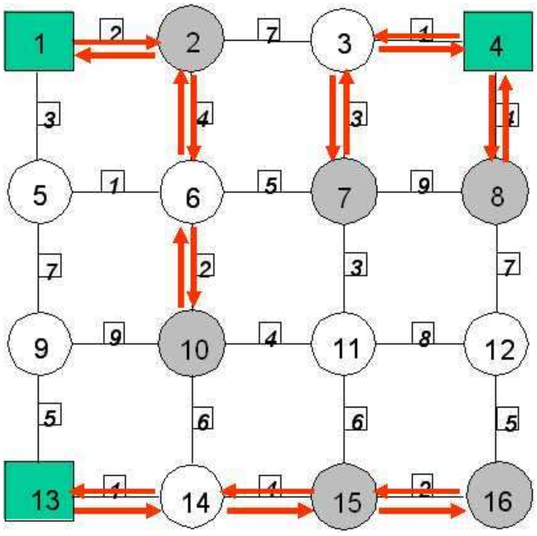

Figure 5.4: The three robot problem; the result given by the algorithm based on the heuristic partition. The three robots start at specified nodes, denoted by squares, travel as directed, and end at the their initial positions.

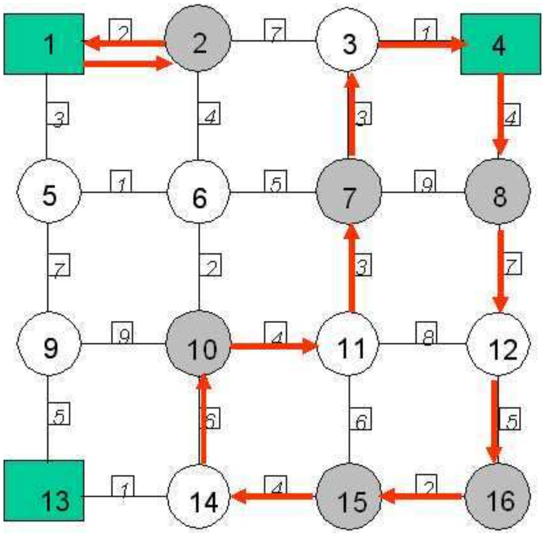

Figure 5.5: The optimal solution of the three robot problem given by the IP formulation.

5.4 Multiple Robots; Fixed Number of Robots with Choice of their Initial Placement

5.4.1 Uniformly Distributed Itineraries

Here we assume that in addition to the model considered in Part II A, we can choose the initial positions of the robots based on a priori information we have about the network. This situation is realistic and may arise for example in lifesaving operations, where the number of automated devices used to search an area is minimized, and supplement the subsequent human effort, eg., fires, earthquakes, avalanches, etc.

Therefore, we want to choose k nodes in the network as initial positions of the k robots, so that this will give us an opportunity to efficiently carry out searches on any given itinerary. Thus, we do not have any a priori information about the itinerary. One natural way to represent this problem is the so-called *k-median problem*. Given n points p_1, p_2, $\ldots$, p_n, we want to choose k of them as centres. Then we form clusters, assigning each of the remaining $n-k$ points to the centre that is closest to it. We want to choose the k centres so that the total distance from the points to the k cluster centres is minimized.

If p_{ij} stands for the j^{th} point assigned to the i^{th} centre, our goal is to

$$\text{minimize} \sum_{i=1}^{k} \sum_{i^{th}\ cluster} \text{dist}(p_i, p_{ij}).$$

When $k \geq 3$, the problem is NP-complete. We will use the approximation algorithm described in [4] The algorithm uses a 3-stage optimization based on LP to achieve a $6\frac{2}{3}$-approximation of the optimal solution, i.e. the k cluster centres selected by this algorithm guarantee a total distance of no more than $6\frac{2}{3}$ times the optimal.

The algorithm runs in polynomial time, dominated by the time necessary to solve the LP instances within. However, we cannot use this algorithm straightforwardly to solve our problem. In our network the distances between pairs of locations might not be known, as generally there will be multiple paths connecting them, each path consisting of more than one edge. It is also possible that there will be no path between a pair of locations. Thus, we have to compute shortest paths between all pairs of nodes in our network and set the length of this shortest path as the distance between these nodes for the purposes of the algorithm [4]. In addition, we have to set the lengths of the non-existing paths to a sufficiently large number, say n times the length of the longest distance in the network. Now we are ready to run the algorithm and we will obtain the k cluster centres. At this point the problem is identical to the one of Part A. Given an itinerary, we have our network subdivided into k clusters. We then identify the itinerary nodes belonging to each cluster, and the robots will visit their respective parts of the itinerary based on shortest path computations done according to the single robot approach described in Part 2.

Formally, the algorithm can be described as follows:

1. Find the shortest path between all pairs of nodes (p_i, p_j) in the network as described in Part 2. Denote these lengths by $\tilde{c}_{ij}$. If no path exists, then we set $\tilde{c}_{ij} = MaxD$, where $MaxD$ is a sufficiently large number.

2. Run the k-median algorithm on the initial network. The algorithm outputs a set of centres $(p_1, p_2, \dots, p_k)$ and for each node in $(p_{k+1}, p_{k+2}, \dots, p_n)$ the assignment to its respective centre.

3. Place the k robots at the centres $(p_1, p_2, \dots, p_k)$ computed in step 2.

4. Given an itinerary $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_m\}$, subdivide into k disjoint itineraries $\mathcal{I} = \tilde{I}_1 \cup \tilde{I}_2 \cup \dots \cup \tilde{I}_k$, such that $\tilde{I}_i \cap \tilde{I}_j = \phi$ for $i \neq j$. The nodes in $\tilde{I}_i$ belong to the cluster of the i^{th} robot.

5. Run the algorithm of Part 2 for robot i for the itinerary $\tilde{I}_i$.

5.4.2 Non-uniformly Distributed Itineraries

In the formulation of the problem in the previous section we assumed that each node is equally likely to become an itinerary node. In the real world we can imagine situations when we a priori know that some locations are more probable to be itinerary locations than others. For example, police patrol checks in a city. Some streets, neighbourhoods, or hot spots are checked more often, i.e. they are more likely to be on the itinerary. The way to incorporate this into our model is to define a probability distribution over the nodes of the network. Equivalently, we can assume that this distribution is normalized and each node has a weight w_i assigned proportional to the probability of this node being an itinerary node. We want to place the robots so that the (yet to be determined) itinerary is carried out efficiently. Again, this can be done by clustering. In the literature this is known as the *k-median problem with centre costs*. The algorithm from [4] can be easily modified to achieve a factor of 10 approximation to the optimal total distance. As one might expect, depending on the weights, the centres might not correspond to ones chosen in the unweighted k-median problem. As clear from this discussion, the only step that will be affected by this is step 2. Of course, we need some preprocessing, namely transforming the probability distribution into weight assignments. This can be done in either step 1 (as it does not affect the shortest path computations) or in a separate step, before running the algorithm for the k-median problem with centre costs.

5.4.3 Simulation Result

We implemented the k-median algorithm, both the unweighted version and weighted one, in Matlab. Figure 5.6 shows the clustering result for the unweighted case. The weights for the weighted network are the same as in the unweighted network (i.e., uniform) except that the weight of node 1 in the weighted network is 10 times the weight of any other node. The clustering result for the weighted network (which is not shown here) differs from that of the unweighted network only in the location of the centre node for the cluster that contains node 1; it moves from node 25 in the unweighted case to node 18 in the weighted case (that is, closer to node 1).

5.5 Multiple Robots; Number of Robots to be Determined

In Problem C, we need to decide the minimal number of robots needed to travel the itinerary such that the total distance travelled by the robots are bounded by the value d (the parameter d is supplied by the user). The strategy we are proposing is to first consider Problem B with i robots (the value of i starts from 1). After solving this problem we denote the total distance travelled by d_i. If d_i is less than d, we know the number of robots needed is i. Otherwise, we consider Problem B with $i + 1$ robots and check whether $d_{i+1} < d$.

Note that this algorithm will not run for ever, because if we assign the same number of robots as the number of itinerary nodes, then we can simply put them on each node and let them not move at all. The total distance travelled is 0 in this case.

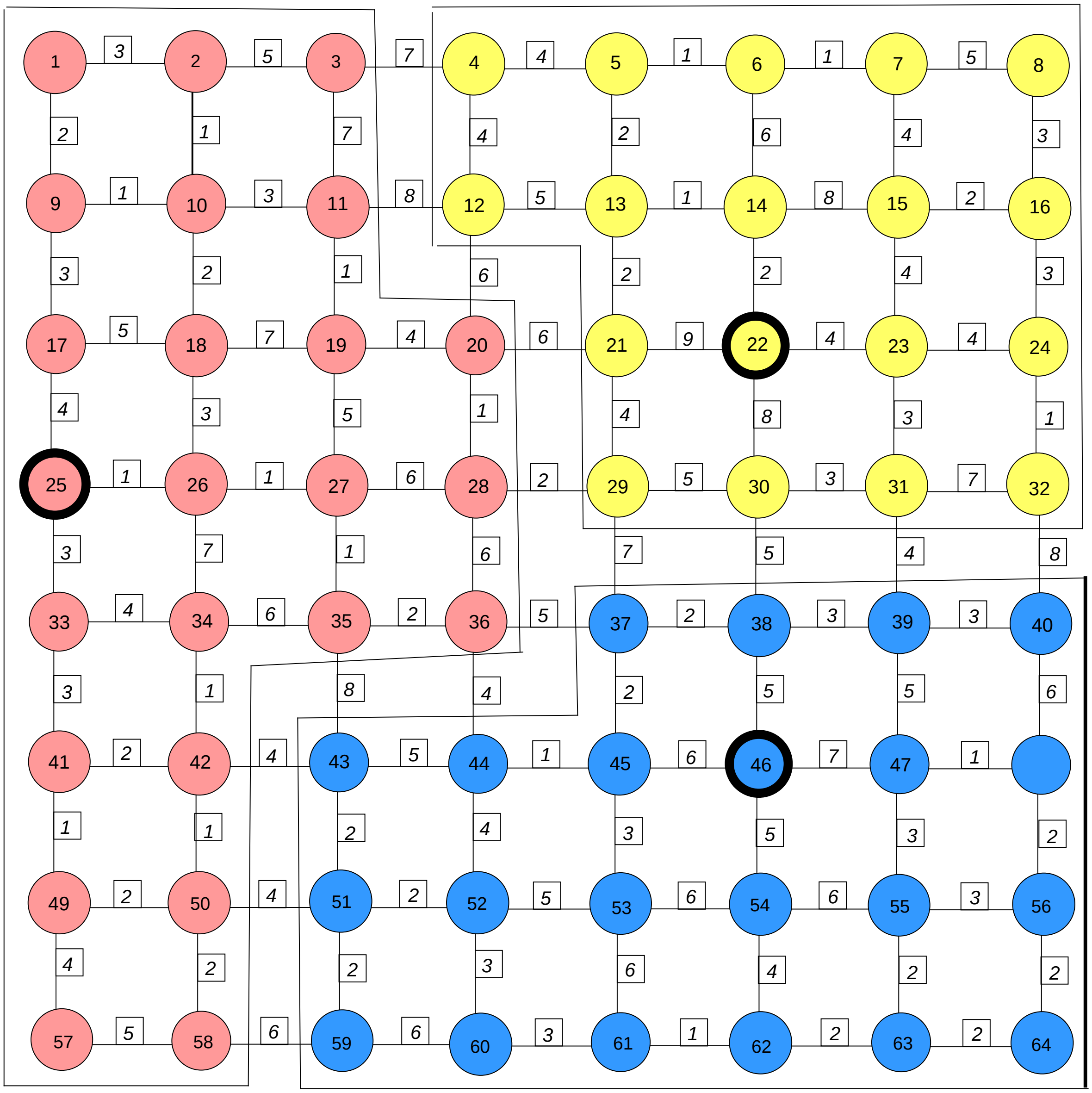

Figure 5.6: Clustering results for the unweighted network given by the k-median algorithm. In the figure the nodes are divided into three clusters, as specified by the bounding boxes, and indicated by the colours pink, yellow, and blue. One robot is assigned to each cluster, starting at the node circled (the ‘centre node’; nodes 25, 22, and 46).

5.6 Random Events

Here we consider again the case of a single robot. However, now the arcs may or may not be usable. This situation may arise, for example, in a city where some routes may be blocked because of traffic or construction. To model this we assign a probability p_{ij} that the arc joining nodes i and j is blocked. The robot will not know whether an arc is blocked or not until it reaches an adjacent node. Further, we assume that the network does not change during the tour of the robot, i.e., we assume a static network. However, the network may change (because of the random events) between itinerary runs.

Our approach is to initially ignore the fact that some arcs may be blocked and run the algorithm described in Part I. We obtain a tour for the robot and then compute the 'risk' r of this tour by computing the probability that at least one arc of the tour is blocked. If r is less than some specified risk threshold R then the tour is taken. If the tour is too risky, $r > R$, then we exclude this tour from the possible paths and re-run Part I to obtain the next best tour. Again we compute the risk. We iterate this procedure until we find a path that is 'safe enough', ie., $r < R$, or else determine that there is not a safe enough tour to carry out the itinerary.

In the case when we find a safe tour, we then compute the 'second order' risk of this path. That is, if one of the arcs happens to be blocked when the robot arrives there, how costly is the detour around this arc to continue carrying out the itinerary? It could happen that a safe path in the sense of $r < R$ may be too risky in the second sense. This second order risk can be interpreted as the 'expected length' of the tour, i.e., if a robot performs that tour many times, on average what is the length the robot travels due to the random broken arcs it encounters?

To estimate the second order risk of a path, we first have to decide on a protocol for the robot to follow if faced with a blocked arc: Follow the shortest path from the current node to the next node in the itinerary. Then we approximate the expected length of the tour by considering only 'first-order' detours; if an arc ij with $p_{ij} > 0$ is blocked, find the length of the tour assuming the robot follows the shortest path from it's current node to the next itinerary node, and furthermore assuming that no other arcs will be blocked. A weight of p_{ij} is assigned to this length. Now average over all such (weighted) detours.

Bibliography

[1] M. Talbot (`talbotm@acm.org`), *A Dynamical Programming Solution for Shortest Path Itineraries in Robotics*, Ryerson University, April 2003.

[2] W. Winston and M. Venkataramanan, *Introduction to Mathematical Programming*, Thompson Brooks/Cole, 2003.

[3] G. Pataki, *Teaching Integer Programming Formulations Using the Travelling Salesman Problem*, SIAM Review, Vol. 45, No. 1, pp. 116–123, 2003.

[4] M. Charikar, S. Gupta, E. Tardos, D. Shmoys, *A Constant-factor Approximation Algorithm for the k-median Problem*, Proceedings of the ACM Symposium on Theory of Computing (STOC ’99), Atlanta, Georgia, USA, 1999.

[5] W.J. Cook and W.H. Cunningham and W.R. Pulleyblank and A. Schrijver, *Combinatorial Optimization*, John Wiley & Sons Inc., New York, 1998.

Chapter 6

Assessment of Stormwater Concentration Data

Participants: Peter Ehlers (Mentor), Edward (Shan) Chang, Song Li, Gabriel Mititica, Beiyan Ou, Yulia Romaniuk, Shijun Song, Xuekui Zhang.

PROBLEM STATEMENT: Stormwater run-off in Calgary affects the water quality of the Bow River. In order to assess the impact on the Bow River, the city has been monitoring pollutant levels in stormwater over the past several years. Seven pollutants have been monitored in four general land use areas in the city—Developing, Residential, Industrial and Commercial areas. City engineers are interested in modelling the pollutant levels in the different areas and in changes over time. They suggest (based on other work in this field) that a lognormal model would be reasonable. Model parameters will have to be estimated. The data provided are for the years 2001 and 2002.

List of Participants

Organising Committee

Rex Westbrook	University of Calgary
Kes Salkauskas	University of Calgary

Local Committee

David Leeming	University of Victoria
Reinhard Illner	University of Victoria
Pauline van den Driessche	University of Victoria
Julie Zhou	University of Victoria

Mentors

C. Sean Bohun	Penn State University
Tim Myers	University of Cape Town
Tobias Schaefer	University of North Carolina at Chapel Hill
Petra Berenbrink	Simon Fraser University
Randall Pyke	University College of the Fraser Valley, Abbotsford, BC
Peter Ehlers	University of Calgary

Students

Pengpeng Wang	Simon Fraser University

Boyan Bejanov	University of Alberta
Robert Junli Liao	University of Alberta
Yury Petrachenko	University of Alberta
Malcolm Roberts	Univeristy of Alberta
Yulia Romaniuk	University of Alberta
Mengzhe Wang	University of Alberta
Mohammed Ali Yassaei	University of Alberta
Wang Zhian	University of Alberta

Wan Chen	University of British Columbia
Gabriel Mititica	University of British Columbia
Shijun Song	University of British Columbia
Xuekui Zhang	University of British Columbia

Nancy Azer	University of Victoria
Edward (Shan) Chang	University of Victoria
Michelle Edwards	University of Victoria
Zhu Jiaping	University of Victoria

Beiyan Ou	University of Victoria
Mahin Salmani	University of Victoria

Karel Casteels	Carleton University
Jason Lobb	Carleton University
Stanislava Peker	Concordia University
Benjamin Chan	Cornell University
Radu Haiduc	Cornell University
Andrei Maxim	Cornell University
Andreas Hofinger	Industrial Mathematics Institute, Linz, Austria
Abramov Vilen	Kent State University
Mohammad Al-Khaleel	McGill University
Bo Zeng	Purdue Univeristy
Lee Jinwoo	Seoul National University
Heejeong Lee	Seoul National University
Yan Wu	University of Manitoba
Joohee Lee	University of North Carolina at Chapel Hill
Song Li	University of Saskatchewan
Tzvetalin Vassilev	University of Saskatchewan
Amirhossein Amiraslani	University of Western Ontario
Hooman Javidnia	University of Western Ontario
Olga Krakovska	University of Western Ontario
Azar Shakoori	University of Western Ontario
Benjamin Akers	University of Wisconsin-Madison

PIMS Contact Information

email: pims@pims.math.ca
http://www.pims.math.ca

- **Director: I. Ekeland**
  Phone: 604-822-3922
  Fax: 604-822-0883
  email: director@pims.math.ca
- **Deputy Director & SFU-Site Director: M. Trummer**
  email: sfu@pims.math.ca
- **UAlberta-Site Director: T. B. Moodie**
  email: ua@pims.math.ca
- **UCalgary-Site Director: G. Chen**
  email: uc@pims.math.ca
- **UVic-Site Director: C. Bose**
  email: uvic@pims.math.ca
- **UWashington-Site Director: G. Uhlmann**
  email: uw@pims.math.ca