Using Optimisation and Genetic Algorithms to Inform Logistics Decisions

Paul Derbyshire
Senior Technical Advisor
BearingPoint

October 2023

GUROBI OPTIMIZATION

BearingPoint

Content

1 Introducing BearingPoint

2 The Challenge

3 Exploring Our Options

4 Crafting the Solution

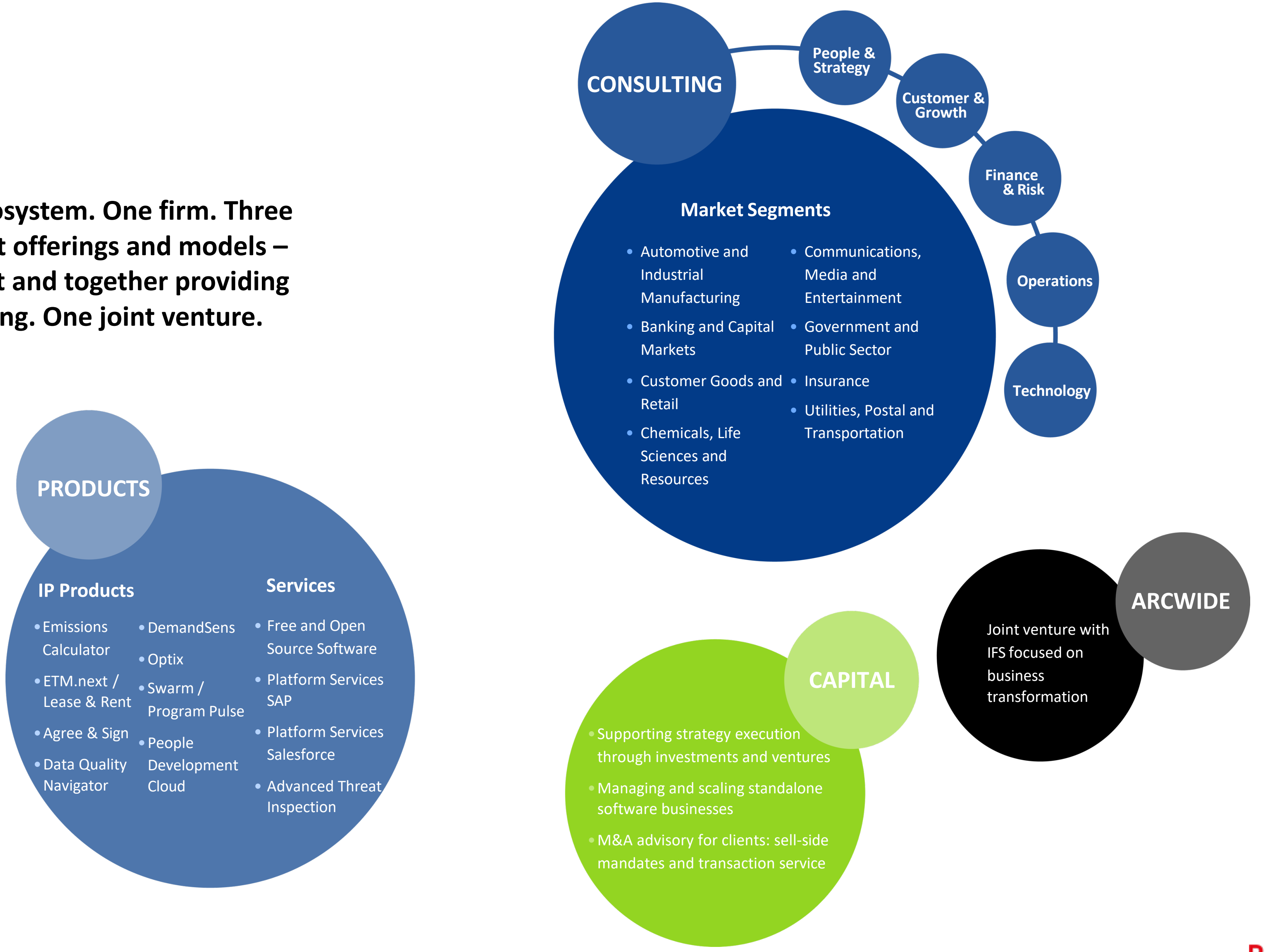
5 Lessons Learned

6 Key Takeaways

BearingPoint®

A story of
unity

The unique BearingPoint ecosystem. One firm. Three business units – with distinct offerings and models – focused on their own market and together providing technology-enabled consulting. One joint venture.



BearingPoint.

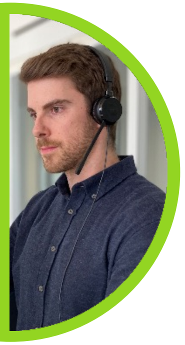
Our year in
numbers

Numbers only tell part of a story, but they show how far we've come. In 2022, we exceeded our own predictions and grew stronger with every step.

BearingPoint people
5,253
(Client facing: 4,657)

23% ↑
People increase year-on-year

206
Firmwide trainings



24
IP products

24
Countries with BearingPoint offices

47
BearingPoint offices

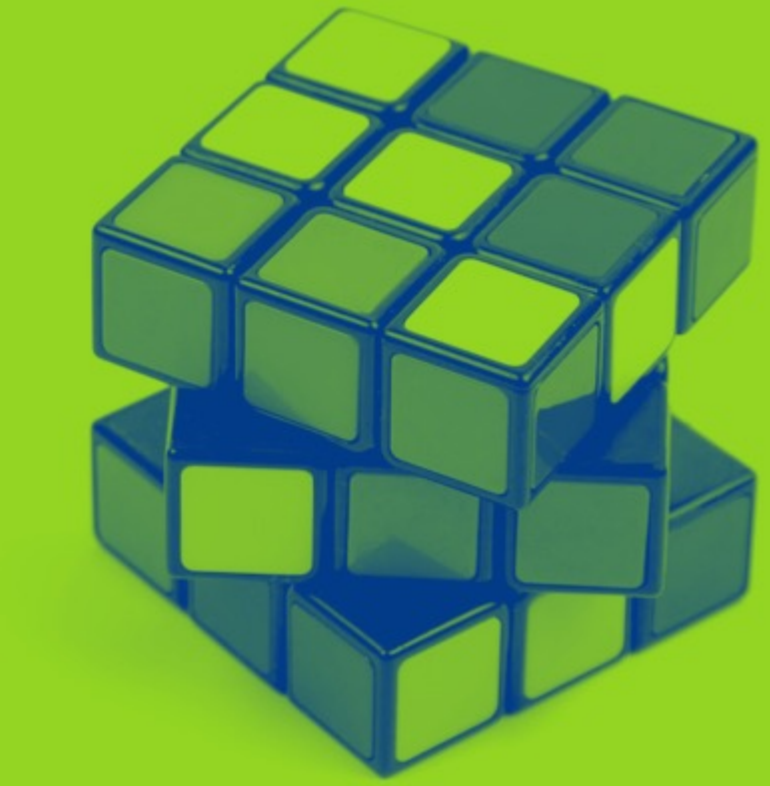
24% ↑
Revenue growth year-on-year
(Adjusted without revenue from Beyond business and member firms)

Total revenue
€863m

Projects delivered
1,603

The Challenge

What were we being asked to solve?



Setting the scene

A major **grocery** and **general merchandise** retailer

Significant **growth**

Multiple product types and **temperature regimes** across the product portfolio

- Manually Handled & Automation Handled Products
- Chill / Fresh
- Frozen
- Ambient
- Temperature sensitive

Designing a **new automated DC** to service stores and provide elements of **hub distribution**

Serving **hundreds of stores**

The Challenge:

Create a schedule that meets store service requirements with optimised fleet and automation usage

Create a schedule that meets store service requirements, optimises fleet and maximises use of automation

Optimising across multiple related functions

Create a cost optimised schedule for key activities for peak and average days. Taking into consideration:

- Order release times
- Store delivery windows and volumes
- Trailer parking and electric hook-up
- Unloading / loading bay capacities

- Fleet size and mix
- Storage capacity
- Automation capacities
- Drive times

- Shift labour
- Delivery and vehicle costs
- Labour costs

Key activities for scheduling

Inbound → Picking → Extraction → Loading → Dispatch → Store Delivery

In order to be effective and to maximise the return on the investment in automation the whole site needs to operate as a single, inter-linked, holistic entity (akin to a manufacturing production plant)

BearingPoint

The size and complexity of the challenge

Creating a feasible schedule within the constraints of the site design is highly complex

- There are many **competing factors to consider and test at the same time**
- Each “decision” **impacts processes and capacities before and after**
- There may be **a number of possible solutions**

Feedback and adjust to meet constraints

Store Arrival Schedule → **Dispatch Schedule** → **Loading Schedule** → **Pick Schedule** → **Inbound Schedule**

Store Arrival Schedule
- Store delivery **constraints**
- **Pallet volumes** of each type
- Number of **separate deliveries** and vehicles

Dispatch Schedule
- Store travel **times**
- **Yard parking** spaces for trailer pooling

Loading Schedule
- **Loading effort**
- **Marshalling**

Pick Schedule
- **Automation pick capacity**
- **Manual picking** requirements
- **Pre-pick storage** space
- **Dispatch door** availability

Inbound Schedule
- **For certain products** inbound schedule is dependent on outbound schedule

- **Hundreds of stores**
- **Multiple store delivery slots**
- **8 product types**
- **Fleet size and mix**
- **Site constraints**
- **Time granularity of 15 mins**

There could be 10 million+ variables in this problem

Exploring Our Options

What options did we look at?

What approaches did we consider?

	The Good	The Less Good
Single MIP Optimisation model	• **Deals with everything in a single MIP** model. Conceptually more straightforward • Does not require additional functions to assemble sub models	• **The size of the problem!** The number of variables and constraints, based on previous examples of MIPs we'd run for similar logistics models
A heuristic backward-scheduling model	• Potentially **quicker** to build, easier to build in specific rules of thumb • Would require **some level of simplification**	• **How do we choose the first delivery schedule and vehicle type?** • The number of variables and rules would still be significant to manage
A backward-scheduling model inside a harness	• Allows us to **evaluate multiple delivery** schedules and vehicle types as starting points for the backward schedule	• **Which type of harness?** Genetic Algorithm? Reinforcement Learning?
A backward-scheduling model inside a Genetic Algorithm harness	• Allows us to **parallelise** the **optimisation** of **multiple delivery schedules** • Ease of use	• **Convergence can be slow** due to chromosome length

Using a chromosome to represent a schedule

A chromosome made up of multiple genes

Delivery Slots *i* - *i+2*			Delivery Slots *i,* - *i+2*			Delivery Slots *i* - *i+2*			Delivery Slots *i* - *i+2*		
1	1	1	1	0	0	1	1	0	1	1	0
Store *j*						Store *j+1*					
Vehicle Type *k*			Vehicle Type *k+1*			Vehicle Type *k*			Vehicle Type *k+1*		

Each gene in the chromosome codifies

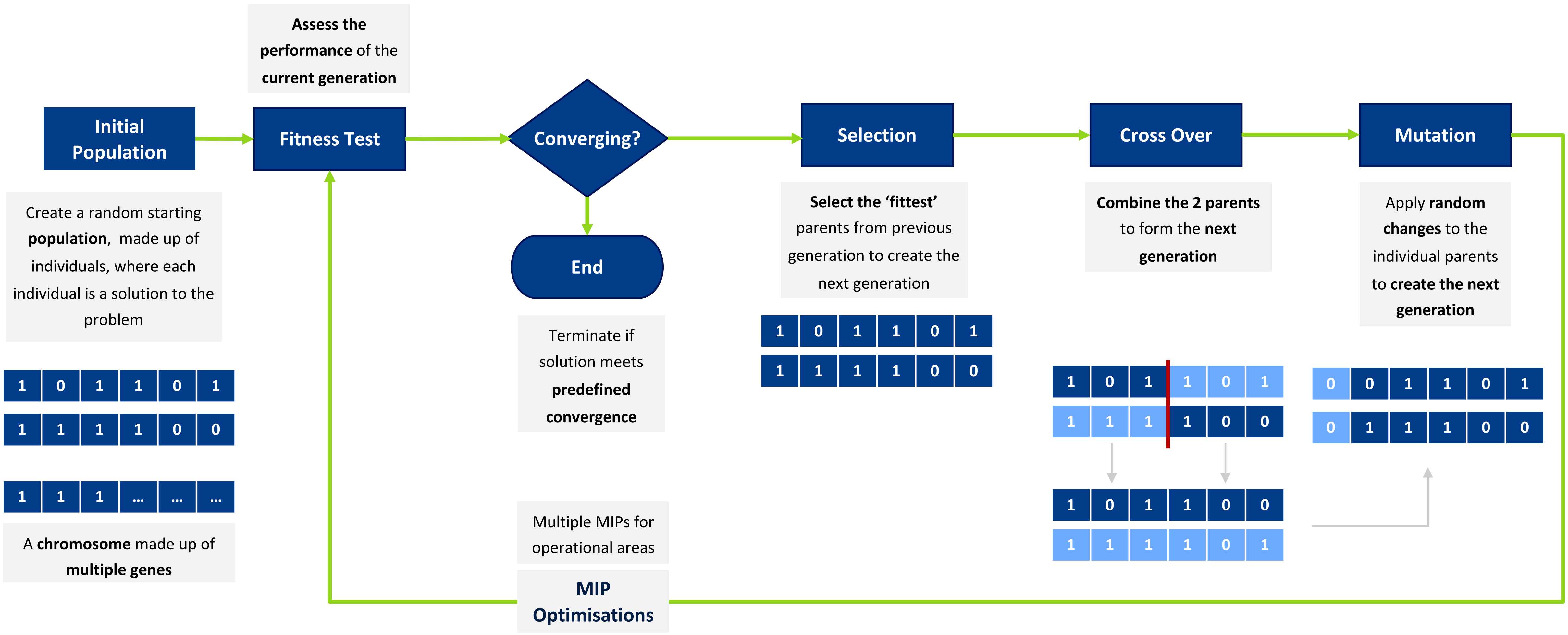
Delivery time

Store number

Vehicle type

An alternative approach using genetic algorithms

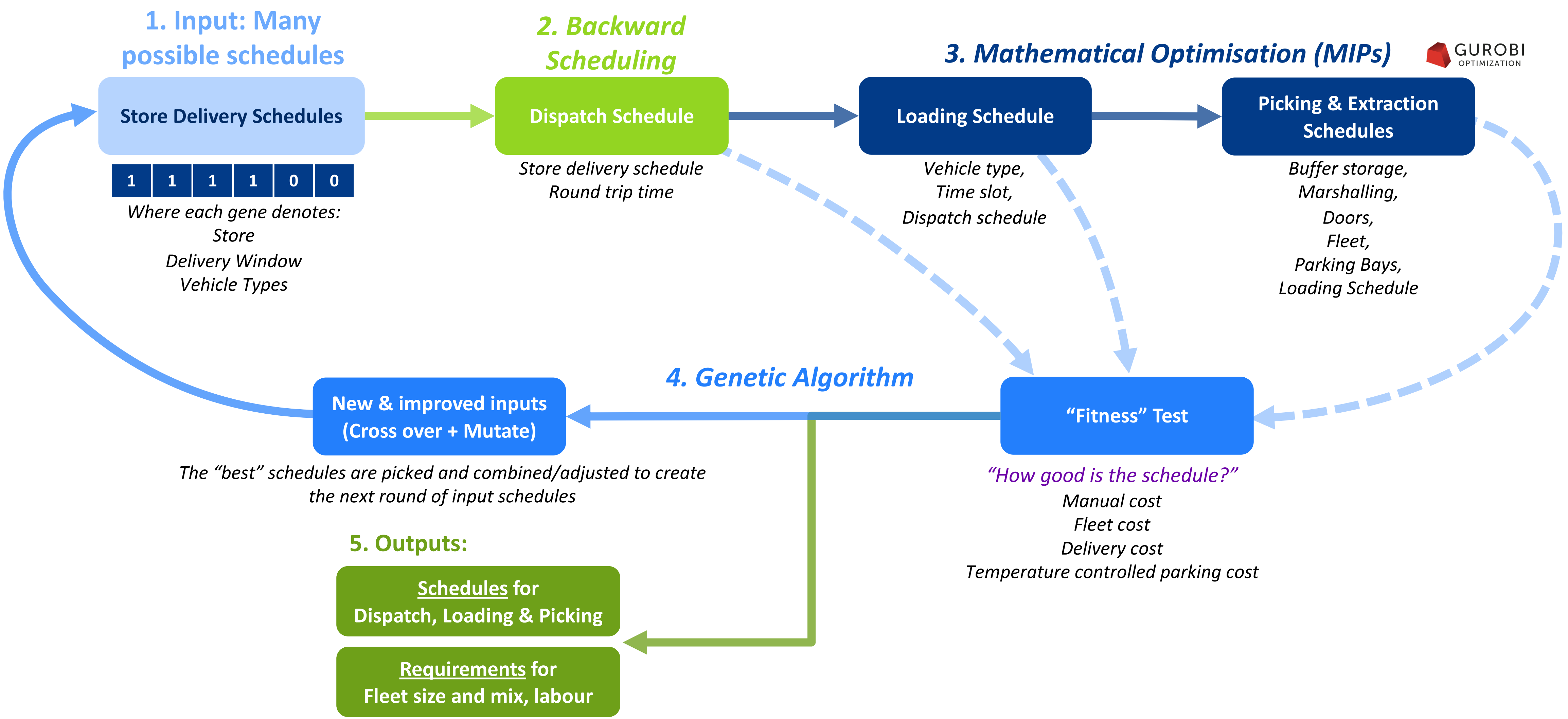
Using nature's way to evolve to the best solution



BearingPoint.

Crafting the Solution

What did this look like for us?



Infrastructure used

Hosting + Genetic Algorithm Harness

Azure to host data and provide parallel processing power for **data pipelines, MIP generation, Gurobi communication and Genetic Algorithm**

MIPs and Genetic Algorithm outputs 'translated' into real world terms for visualisation

All code written in Python

LP File Generation

Pyomo used to create the LP file for Gurobi (and open source one solvers that we did basic testing with)

Optimisation

Gurobi used for all MIPs

Multiple Gurobi configurations assessed

MIPs written directly to Gurobi using the **Gurobi API**

Currently running on Gurobi Cloud

Visualisation

Power BI

A number of formats of outputs were provided

The main reporting element is provided by Power BI

Used for drill down to 'understand' and explore answers – for validation and presentation

Some key facts

Chromosome length

16k-64k

Chromosome population

96

Generations Run

Up to

20

… before costs convergence

MIPs used per Chromosome

5-14

To cover functional areas and product types

Speed

Up to 33 MIPs per minute

16 Chromosomes in parallel

MIP binary variables

~800k

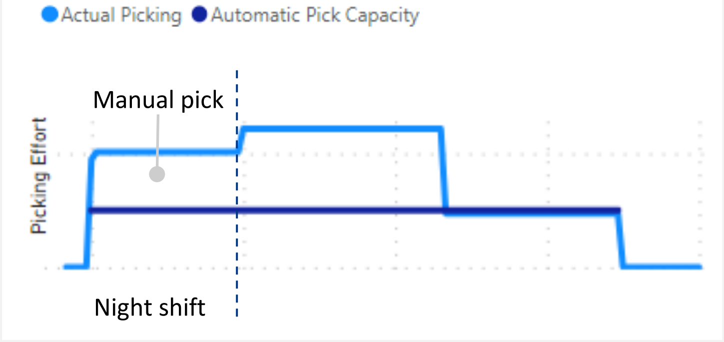
MIP constraints

~30k

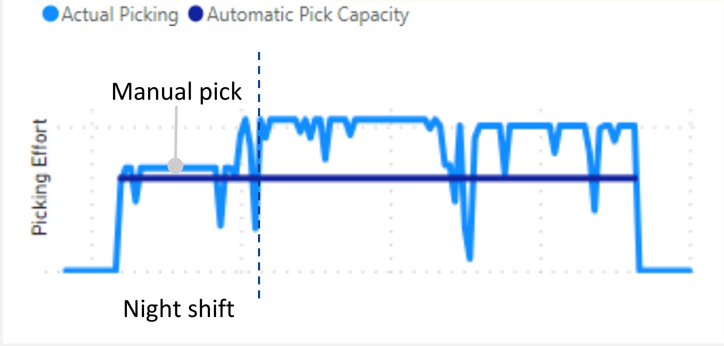
Sample scenario modelling

Scenario Outputs

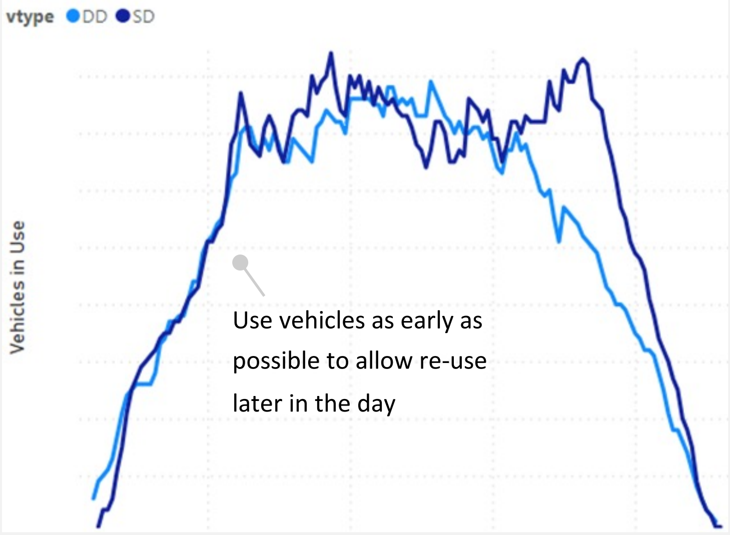
Chilled Picking | Window: 00:00 – 21:00



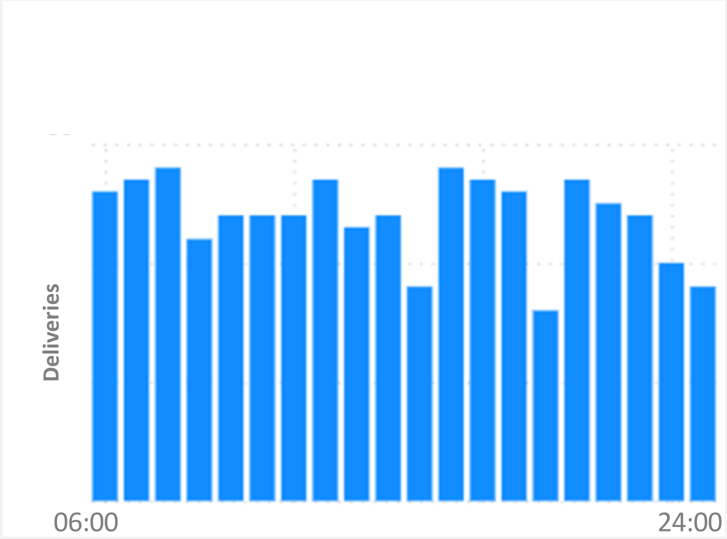
Ambient Picking | Window: 01:00 – 22:00



Fleet Utilisation | 24 Hour Window



Deliveries | 24 Hour Window



Key Outputs

- Optimal timings for each activity required to get all orders to stores
- The number of colleagues required on each shift to complete work
- Optimal delivery vehicle type mix and number required
- Identified potential future peak day bottleneck areas
- Clear indication of how store delivery timings impact DC activity and DC labour requirements

Key Takeaways

Lessons learned

- **Optimising the end-to-end** operation in small time periods **is complex**
- **Genetic algorithms are a great way to parallelise an optimisation(s)**
- A single **complex MIP** can be **traded for multiple MIPs** with a Genetic Algorithm but at the **cost of convergence** time
- Iterative building and testing worked well. **Start with shorter chromosomes and build up from there**
- There's always a way to get to the answer!

Key takeaways

3 key themes

Genetic Algorithms

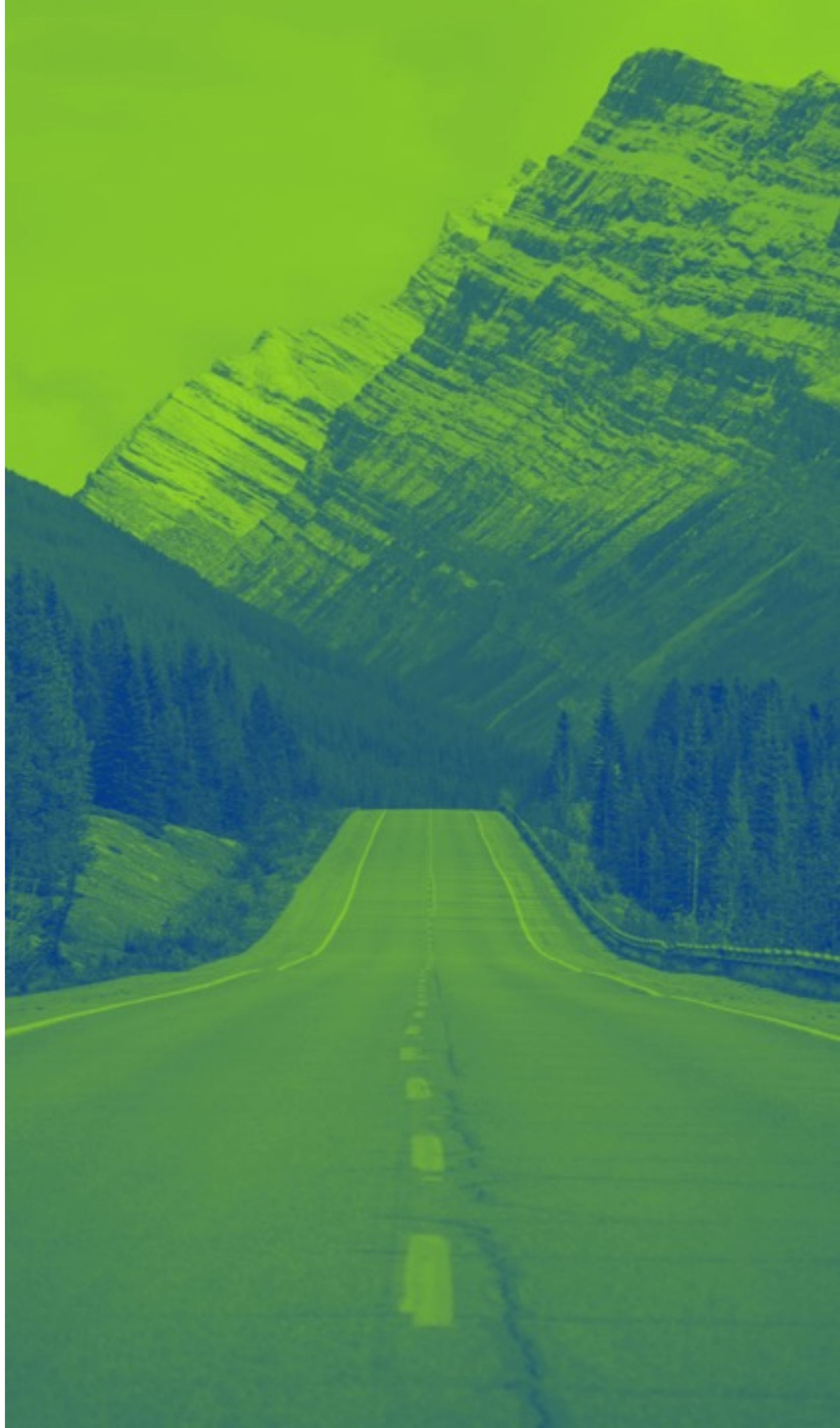
Are a great approach to solving large scale problems. Combining with MIP can speed up the evolution

Break the problem down

Assess ways in which functional areas can be optimised and combined

Parallel running

Make use of the parallel run capability afforded by Genetic Algorithms



BearingPoint®