**Discrete Optimisation Project Documentation**

# Info

For complex constraints a line explaining what the constraint is controlling should be placed above. *[An additional detailed note about how exactly if works can be placed above the constraint in square brackets and italics if needed, keep this to a minimum]*

Please just steal maths from the file “DODM\_07\_ILP-VehicleRoutingProblems.pdf” and leave comments about what you’re stuck on [Review Tab > New Comment]

# Documentation Notes

* The commodities that a person must track are:
  + money, number of times changed public transportation, spending costs, current mode of transportation, health
  + We will have to nick (steal) constraints from commodity flow maybe? I would just like to sum out where they go i.e. [sum(places been \* cost of places) < total money] but I’m not sure it’s that simple
* The assumption about a person can’t return to a node they have previously travelled too must be confirmed
* The capacity of nodes and modes of transports is not yet considered by the model

# Assumptions

* A person will not return to a node they have personally travelled to (this needs to be confirmed)
* A bus spends no time at a bus stop
* A bus can be caught anytime within a x second window before it arrives at a point. This time is called the bus relaxation (I assume 60 seconds is fine, however we can tune this)
* The travel times of two people riding the same bus are modelled separately and may lag between each other according to the bus relaxation parameter
* The indexing of moves is currently assumed to be {1 = walking, 2 = Cycling, 3 = Bus}. I will be worth asking the prof if we are ok to always assume this is the case. Otherwise, there will have to be some hardcoding in the model to catch instances in the inputs where this is not the case.
* The bike station capacity is checked at x min intervals (default 30 mins), there will have to be a consultation about how this linear relaxation could be best handled

# Indexes

# Constants

# Independent Variables

# Semi-Dependant Variables

***These are the variables that are technically dependant variables but are modelled as constrained independent variables***

# Dependent Variables

***This section is saved for any variable which are fully dependant on other variables, this is generally used if various variables need to be consolidated into a single figure i.e., the sum of costs. Please note that when using these variables, the developer needs to be careful not to make any illegal calculations this new variable***

# Constraints

## Basic Conservation of Flow

Flow is directional, multi-medium, multi-flow (person)

*[the entries/exits from a node j, needs to be larger than one if there is a task at the node or if it is the home node of the person (h) (Boolean values)]*

Currently each node can only be visited once to not interfere with constraints around timing

## Task Timing

This controls the time (w) which the task at j starts

This controls that tasks happen within their designated time windows (only applies if the tasks happens and the task extension penalty for falling out of the allotted time isn’t applied)

Task extensions/delays only apply to tasks that happen

## Bus Travel Constraints

These two constants state that person must finish their task and waiting period at node i, x seconds (controlled by the bus relaxation constant) before the exact bus they want to catch arrives at

The person can only get bus (i, d, r) the relevant Bus\_idr must be positive

Every time a person gets on a bus from another mode of transport, they must purchase a bus fare

## Bike Capacity Constraints

The bike station capacity is checked at x min intervals (default 30 mins), there will have to be a consultation about how this linear relaxation could be best handled

At the end of the period, there must be a non-negative quantity of bikes and free spaces at each bike station

## Unfinished Tasks Penalties

A penalty Qt must be paid for a task t ∈ Tn not performed