

Technical report 12-015

Micro-ferry scheduling problem with time windows*

M. Burger, B. De Schutter, and J. Hellendoorn

If you want to cite this report, please use the following reference instead:

M. Burger, B. De Schutter, and J. Hellendoorn, "Micro-ferry scheduling problem with time windows," *Proceedings of the 2012 American Control Conference*, Montréal, Canada, pp. 3998–4003, June 2012.

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.51.19 (secretary)
fax: +31-15-278.66.79
URL: <http://www.dcsc.tudelft.nl>

*This report can also be downloaded via http://pub.deschutter.info/abs/12_015.html

Micro-Ferry Scheduling Problem with Time Windows

M. Burger, B. De Schutter and J. Hellendoorn

Abstract— We propose a method to solve scheduling problems taking into account energy-efficiency and variable speeds. We focus on a scheduling problem for autonomous micro-ferries, where the task of assigning transportation requests to specific micro-ferries and determining the order of handling them is done centrally. The method is based on the travelling salesman problem and vehicle routing problem with time-windows, but differs in the inclusion of constraints on the energy-consumption and an increased flexibility in travel times, which are both influenced by varying the speed of the micro-ferries. This results in a mixed-integer non-linear programming problem, which can be transformed into a mixed-integer linear programming problem by using an approximation of the speed variables.

I. INTRODUCTION

The Port of Rotterdam in the Netherlands stretches from the North Sea up to the centre of the city. Currently, it is being expanded at the sea side by reclaiming 16 km² of land from the sea [1]. Due to the expected movement of the port industry from the *City Ports (Stadshavens)* to this new area *Maasvlakte 2*, the harbour area close to the city centre becomes available for development of new living and business areas. In order to connect the different parts of the City Ports, transportation over water is expected to become important. Envisioned is a water bus network called *Aquanet*, which will be manned and operated with a fixed schedule. Besides this water bus network there would be possibilities for an *on demand* autonomous micro-ferry network to account for fast and personal transportation between several locations in the harbour. Figure 1 shows a schematic view of the Rotterdam City Ports, including an example network of docking locations for the micro-ferries. In this paper we describe the modelling of such a micro-ferry network, and develop the optimisation problem that needs to be solved to schedule transportation requests, taking into account both the possibility to travel at different speeds and the energy consumption of the micro-ferries.

The problem of scheduling transportation requests in a harbour using micro-ferries is closely related to several standard optimisation problems in logistics, such as the pick-up and delivery problem [2], [3], the travelling salesman problem [4], the vehicle routing problem [5], and the dial-a-ride problem [6], but differs in the use of variable speeds, and thereby variable costs in the objective function. We want to optimise the transportation between a fixed number of

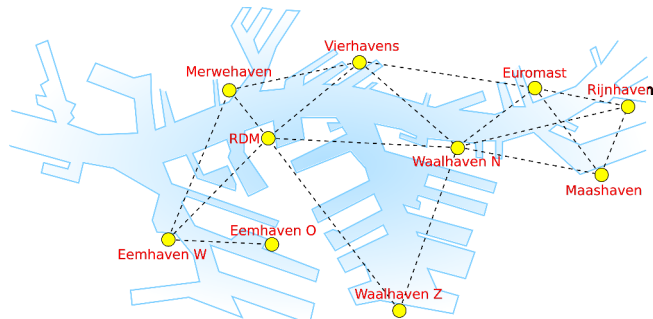


Fig. 1. Schematic view of the City Ports with an example network of pick-up and delivery locations

stations, where a small group of people can share a ferry, just like a taxi on land. The ferries will pick the people up at one station, and directly deliver them to their desired destination; we assume that no stops are made in between to add more people to the ferry. As such this problem becomes a dial-a-ride problem, where first all passengers have to be delivered at their destination, before a new pick-up can be made. We solve a static problem, as opposed to dynamic problems where e.g. the arrival times of customers are based on random distributions [7].

The novelty in the proposed micro-ferry scheduling problem is the use of variable costs dependent on speeds. Both the energy consumption and the travel times of a micro-ferry will be dependent on the speed, and by assigning a separate speed for each transportation request we add more flexibility to the scheduling problem. By adding constraints on the energy consumption of each micro-ferry, we obtain a schedule that does not assign more requests to a micro-ferry than it can handle based on its energy level.

The structure of this paper is as follows. In Section II the micro-ferry scheduling problem is presented, by introducing the necessary variables and constants. They are used in Section III to introduce the optimisation problem, by stating the objectives and constraints of the system. This leads to a non-linear optimisation problem, for which a linear programming approximation is derived in Section IV. Conclusions and ideas about future work can be found in Section V.

II. PROBLEM FORMULATION

To describe and solve the micro-ferry scheduling problem, we consider two distinct networks. First the physical network is introduced, in which the micro-ferries move between several locations. Afterwards, the problem of scheduling requests is defined as a network problem, where each request corresponds to a node that should be visited once.

This work is supported by the Delft Research Center Next Generation Infrastructures (DRC-NGI), and the European Union Seventh Framework Programme [FP7/2007-2013] under grant agreement no. 257462 HYCON2 Network of Excellence.

All authors are with the Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands. {m.burger, b.deschutter, j.hellendoorn}@tudelft.nl

A. Description of the physical network

We consider a harbour, lake or river where a fleet of M micro-ferries can pick up and deliver customers at L distinct locations. The amount of transportation requests between the locations is denoted by R . The micro-ferries can travel at different speeds bounded by the interval $[\underline{u}, \bar{u}]$, with $0 < \underline{u} < \bar{u}$. The locations are represented by the set $\mathcal{L} = \{1, \dots, L\}$. The matrix $\mathbf{L} \in \mathbb{R}_+^{L \times L}$ contains the path lengths $l_{pq} \geq 0$ between the locations p and q (with $p, q \in \mathcal{L}$); we have $l_{pq} = 0$ if and only if $p = q$.

The customers can make transportation requests to be brought from one location to another. The set $\mathcal{R} = \{1, \dots, R\}$ denotes the different requests; each request $r \in \mathcal{R}$ has a pick-up location $p_r \in \mathcal{L}$, a delivery location $q_r \in \mathcal{L}$, and a desired time-interval $[t_{a,r}, t_{b,r}]$ for the pick-up to take place.

The set \mathcal{R} consists of two types of requests: *current requests* and *future requests*. The set $\mathcal{M} = \{1, \dots, M\}$ denotes the current requests, consisting of the requests that the M micro-ferries are handling at the moment the scheduling problem is to be solved. The set $\mathcal{N} = \{M+1, \dots, R\}$ denotes the future request, which still need to be scheduled in time, and assigned to the micro-ferries. The set \mathcal{R} is defined as

$$\mathcal{R} := \mathcal{M} \cup \mathcal{N} = \{1, \dots, M, M+1, \dots, R\}, \quad (1)$$

with $R = M + N$ the total number of requests.

B. Description of the scheduling problem

The scheduling problem associated with the network described above consists of finding assignments of requests to micro-ferries such that

- i) each requests is handled by one (and only one) vehicle;
- ii) the energy consumption needed to fulfil the requests does not exceed the available energy level;
- iii) the distance travelled by the ferries is minimised;
- iv) the pick-ups for the requests should (preferably) be within the desired time-interval.

The problem can be represented by a graph $\mathcal{G} = (\mathcal{R}, \mathcal{A})$ where $\mathcal{R} = \{1, \dots, R\}$ is a set of nodes associated with the requests, and $\mathcal{A} = \{(i, j) : i, j \in \mathcal{R}\}$ is a set of arcs connecting the nodes. The nodes consist of two groups, one associated with the M current requests and one associated with the N future requests, numbered as defined in the set \mathcal{R} in (1).

1) *Node properties*: Associated with each node $r \in \mathcal{R}$ are variables $t_r \in \mathbb{R}$ indicating the scheduled starting time (the time at which the customer is picked up), $k_r \in \mathcal{M}$ indicating the micro-ferry number and $e_r \in \mathbb{R}_+$ indicating the energy level of the micro-ferry after completion of request r .

We define a cost c_{rr} indicating the distance from the pick-up location p_r to the delivery location q_r of request r as

$$c_{rr} := l_{p_r, q_r}. \quad (2)$$

If the node number $r > M$, it is associated with a future request $n = r - M$, and the distance $c_{rr} > 0$ equals the distance from p_r to q_r . When the node number $r \leq M$, it is associated with a micro-ferry $m = r$ performing a request, and $c_{rr} \geq 0$ equals the distance of the currently handled request; if the ferry waits at a location the distance is zero.

2) *Arc properties*: Associated with each arc $a \in \mathcal{A}$ are binary variables $x_{ij} \in \{0, 1\}$ indicating whether ($x_{ij} = 1$) or not ($x_{ij} = 0$) node j is ‘visited’ directly after node i by a micro-ferry, and constants $c_{ij} \in \mathbb{R}_+$ indicating the ‘cost’ to schedule request j after request i . This cost is defined as the distance needed to travel from the delivery location q_i of request i towards the pick-up location p_j of request j ; when request j directly succeeds request i , the micro-ferry has to travel without a passenger aboard for a distance

$$c_{ij} := l_{q_i p_j}. \quad (3)$$

If the locations are the same, we have $c_{ij} = 0$.

III. DEFINITION OF THE OPTIMISATION PROBLEM

Using the network definitions and the variables described in the previous section, we can define the optimisation problem of this paper. First we describe several objective functions, which define the terms we wish to minimise. After that the constraints on the optimisation variables are given.

A. Objective function

The objective function used in this paper establishes a trade-off between *energy consumption*, *empty-travel distance*, and the *travel time* for customers. A trade-off can be made between the different objectives by using the weighting variables $\alpha_{ec}, \alpha_{et}, \alpha_{tt} \geq 0$ in the objective function

$$J = \alpha_{ec} J_{ec} + \alpha_{et} J_{et} + \alpha_{tt} J_{tt}. \quad (4)$$

Definitions of the objectives J_{ec} , J_{et} and J_{tt} are given next.

1) *Energy consumption*: A common way to model the dynamics of a vessel is by the vectorial representation [8]

$$\mathbf{M}\dot{\boldsymbol{\nu}} + \mathbf{C}\boldsymbol{\nu} + \mathbf{D}\boldsymbol{\nu} + \boldsymbol{\tau}_e = \boldsymbol{\tau}_c, \quad (5)$$

where $\boldsymbol{\nu} = [u, v, r]^T$ is the velocity vector consisting of the surge speed u , the sway speed v , and the rotational speed r , matrix \mathbf{M} is a symmetric, positive definite mass matrix, \mathbf{C} is a skew-symmetric Coriolis and centripetal forces matrix, \mathbf{D} is a symmetric, positive definite damping matrix, $\boldsymbol{\tau}_e$ is a force vector representing external disturbances (e.g. wind and currents), and $\boldsymbol{\tau}_c$ is the control vector representing the forces exerted by the actuators. Using the force balance (5) we can write the kinetic energy of a surface vessel as

$$E_{\text{kin}} = \frac{1}{2} \boldsymbol{\nu}^T \mathbf{M} \boldsymbol{\nu}, \quad (6)$$

and the associated power (due to movement) becomes

$$P_{\text{kin}} = \frac{d}{dt} E_{\text{kin}} = \frac{1}{2} [\dot{\boldsymbol{\nu}}^T \mathbf{M} \boldsymbol{\nu} + \boldsymbol{\nu}^T \mathbf{M} \dot{\boldsymbol{\nu}}] = \boldsymbol{\nu}^T \mathbf{M} \dot{\boldsymbol{\nu}} \\ = \boldsymbol{\nu}^T [-\mathbf{D}\boldsymbol{\nu} + \boldsymbol{\tau}_c - \boldsymbol{\tau}_e] = [\boldsymbol{\tau}_c - \boldsymbol{\tau}_e]^T \boldsymbol{\nu} - \boldsymbol{\nu}^T \mathbf{D} \boldsymbol{\nu}. \quad (7)$$

In order to take the energy consumption of the micro-ferries into account, we use a simplified expression for the power based on the along-path speed u_j for request j . Furthermore, we assume we have a homogeneous fleet of micro-ferries, meaning that all micro-ferries have the same properties. Besides the quadratic and linear terms of (7) due

to the kinetic energy, we also add a constant term to include energy losses due to a running motor when the micro-ferries are not moving. Therefore, the power of the micro-ferries will be a second order function in the speed, written as

$$P(u_j) = p_2 u_j^2 + p_1 u_j + p_0, \quad (8)$$

where the constants $p_0, p_1, p_2 \geq 0$ are properties of the specific micro-ferry model.

We will assign a *constant* speed u_j per request j within an allowed range $[\underline{u}, \bar{u}]$, hence we can obtain the energy consumption associated with this request by multiplying the power by the duration of the request. The time $T_{ij}(u_j)$ associated with a request j can be found by dividing the distance by the speed. Therefore, the energy consumption ϵ_{ij} of request j when it is preceded by request i is given by

$$\begin{aligned} \epsilon_{ij} &= P(u_j)T_{ij}(u_j) = (p_2 u_j^2 + p_1 u_j + p_0) \frac{c_{ij} + c_{jj}}{u_j} \\ &= (p_2 u_j + p_1 + p_0 \frac{1}{u_j}) C_{ij}, \end{aligned} \quad (9)$$

where c_{ij} (defined in (3)) is the path length from the delivery location of request i towards the pick-up location of request j , and c_{jj} (defined in (2)) is the path length between the pick-up and delivery location of request j . Variable

$$C_{ij} := c_{ij} + c_{jj} \quad (10)$$

is a constant representing the total distance travelled when request i precedes request j .

The energy consumption ϵ_{ij} represents the energy that *would be used* when the request associated with node j directly succeeds the request of node i ; if this is not the case, the energy consumption ϵ_{ij} is not actually consumed. Therefore, the total energy consumption can be written as

$$J_{ec} = \sum_{j=1}^R \epsilon_j = \sum_{j=1}^R \sum_{i=1}^R (p_2 u_j + p_1 + p_0 \frac{1}{u_j}) C_{ij} x_{ij} \quad (11)$$

where the energy consumption term

$$\epsilon_j := \sum_{i=1}^R \epsilon_{ij} x_{ij} = (p_2 u_j + p_1 + p_0 \frac{1}{u_j}) \sum_{i=1}^R C_{ij} x_{ij}, \quad (12)$$

represents the energy consumption that *will be used* for request j . A lower energy consumption means lower (fuel) costs for the owner, and less pollution.

2) *Empty-travel distance*: The distances a micro-ferry is travelling without a customer aboard are undesired costs for the owner. Although it is penalised by (11) already, one might want to penalise it more to reduce operational costs.

The empty-travel distance between two requests associated with nodes i and j are given by the costs c_{ij} ; the empty-travel distance of the fleet is found by summing up the costs

$$J_{et} = \sum_{i=1}^R \sum_{j=1}^R c_{ij} x_{ij}. \quad (13)$$

3) *Travel time*: The travel time for a passenger is given by the length of his/her trip divided by the speed, that is

$$J_{tt} = \sum_{i=1}^R \frac{c_{ii}}{u_i}. \quad (14)$$

A lower travel time means a better service for the customer, since they will arrive at their desired location earlier.

B. Constraints

There are several constraints on the optimisation variables of the micro-ferry scheduling problem that need to be satisfied to obtain a useful solution to our problem. These constraints are discussed in detail next.

1) *Scheduling variables x_{ij}* : The variables x_{ij} represent the order of handling the requests; if $x_{ij} = 1$ request j is handled directly after request i by the same micro-ferry $k_i \in \mathcal{M}$. To ensure that all requests are handled by *one and only one* ferry, we use the constraints (see e.g. [4], [9])

$$\sum_{i=1}^R x_{ij} = 1 \quad \forall j \in \mathcal{R}, \quad (15a)$$

$$\sum_{j=1}^R x_{ij} = 1 \quad \forall i \in \mathcal{R}. \quad (15b)$$

The constraints (15a) ensure that all the nodes in the graph \mathcal{G} have exactly one outgoing arc; every request is preceded by exactly one other request. The constraints (15b) ensure that all the nodes in the graph \mathcal{G} have exactly one incoming arc; every request is succeeded by exactly one other request.

2) *Start time variables t_i* : At the time we run the optimisation algorithm, it is likely that some micro-ferries are currently handling a request. To obtain correct start times for the following requests, the start times of the requests that are currently handled are assigned to the vehicle nodes as

$$t_i = t_{0,i} \quad \forall i \in \mathcal{M}, \quad (16)$$

where $t_{0,i}$ represents the start time of the request currently handled by micro-ferry $k_i \in \mathcal{M}$; if no request is handled by micro-ferry k_i one should assign the current time.

The start times t_i of requests $i \in \mathcal{R}$ should be consistent; that is, if request j directly succeeds request i using the same micro-ferry, start time t_j should be *at least* the start time of request i , plus the time it takes to perform the pick-up and delivery of request i , plus the time it takes to move the micro-ferry from the delivery location of request i to the pick-up location of request j . This can be stated as the constraints

$$t_j \geq t_i + \frac{c_{ii}}{u_i} + \frac{c_{ij}}{u_j} \quad \text{if } x_{ij} = 1, \quad (17)$$

for all $i, j \in \mathcal{R}$, or equivalently

$$(t_i - t_j + \frac{c_{ii}}{u_i} + \frac{c_{ij}}{u_j}) x_{ij} \leq 0. \quad (18)$$

This non-linear inequality can be rewritten in an equivalent linear form in two steps. First we substitute the speed variables u_i by their reciprocals

$$w_i := \frac{1}{u_i} \quad \Rightarrow \quad \underline{w} := \frac{1}{\bar{u}} \leq w_i \leq \bar{w} := \frac{1}{\underline{u}}. \quad (19)$$

Secondly we define a large constant T (based on the big-M method [10]). The non-linear inequality constraints (18) can then be substituted by the linear inequality constraints

$$t_i - t_j + c_{ii}w_i + c_{ij}w_j + Tx_{ij} \leq T \quad \forall i, j \in \mathcal{R}. \quad (20)$$

3) *Slack variables $s_{a,i}$ and $s_{b,i}$* : As stated in the problem formulation, a desired time interval $[t_{a,i}, t_{b,i}]$ is associated with each request $i \in \mathcal{R}$ for the start time t_i . Inequality constraints can be used to force the start time to lie within this interval, but this makes it possible for the optimisation problem to become infeasible. To avoid infeasibility we add two slack variables $s_{a,i}$ and $s_{b,i}$ representing the amount of time the start time is scheduled too early and too late respectively. The associated inequality constraints then become

$$t_{a,i} - s_{a,i} \leq t_i \quad \forall i \in \mathcal{R}, \quad (21a)$$

$$t_{b,i} + s_{b,i} \geq t_i \quad \forall i \in \mathcal{R}, \quad (21b)$$

$$s_{a,i}, s_{b,i} \geq 0 \quad \forall i \in \mathcal{R}. \quad (21c)$$

The start times should preferably be inside or close to the time windows. Deviations from the time-windows can be penalised by using the objective function

$$J_{sv} = \sum_{i=1}^R \alpha_{sa} s_{a,i} + \alpha_{sb} s_{b,i}, \quad (22)$$

where the coefficients $\alpha_{sa}, \alpha_{sb} \geq 0$ can be used to alter the relative importance of early or late starting times; if $\alpha_{sa} > \alpha_{sb}$ starting earlier than the desired time interval is penalised more than starting later than the desired time interval.

4) *Assignment variables k_i* : The constraints of (15) assure that each node in graph \mathcal{G} has exactly one incoming arc and one outgoing arc. For scheduling purposes this is not enough. There are two possible situations that need to be avoided.

If there exists a *sub-tour* within the request nodes, it means that the associated requests are not assigned to a micro-ferry. This problem has been addressed and solved by Miller, Tucker and Zemlin in [11] (and extended and improved in [12]), and the solution is known as the MTZ sub-tour elimination constraints. The method is based on the idea of associating potentials to the nodes in the network, and ensuring that the potential increases along the arcs. Here the start times have taken over the role of the node potentials, and (20) can be seen as the sub-tour elimination constraints.

Since we assign a separate node to each micro-ferry, tours that include more than one micro-ferry node can exist. To avoid this, we introduce a method that can be considered to be the dual of the MTZ sub-tour elimination constraints: with every node we associate a current flowing through the arcs. By assigning a unique current to each of the micro-ferry nodes (these nodes can be thought of as current sources), and having the knowledge that all nodes have exactly one incoming and one outgoing arc (due to (15)), we can exclude the possibility that two micro-ferry nodes share the same tour. Indeed, by assuring that node j has the same current as node i if there exists an arc from i to j ($x_{ij} = 1$), there will be a conflict when node j represents a second micro-ferry,

since it already has another current assigned to it. We set

$$k_i = i \quad \forall i \in \mathcal{M} \quad (23)$$

to assign a unique current to the micro-ferry nodes, and

$$(k_i - k_j)x_{ij} = 0 \quad \forall i, j \in \mathcal{R} \quad (24)$$

to assign currents to the other nodes. For a feasible solution the currents k_i in the network have values $1 \leq k_i \leq M$ due to the assignment in (23), and therefore $k_i - k_j \leq M - 1 =: M \quad \forall i, j \in \mathcal{R}$. We can substitute the non-linear constraints (24) by the equivalent linear constraints

$$k_i - k_j + M(x_{ij} - 1) \leq 0 \quad \forall i, j \in \mathcal{R}. \quad (25)$$

This would result in R^2 inequality constraints. We can reduce this to $\frac{1}{2}R(R+1)$ inequality constraints by using

$$k_i - k_j + M(x_{ij} + x_{ji} - 1) \leq 0 \quad \forall i, j \in \mathcal{R}, i \leq j. \quad (26)$$

Note: Inequality constraints (26) eliminate loops in the network. This means that only solutions will be found where all micro-ferries will handle at least one request. This might be unnecessary and suboptimal, and would even lead to infeasibility of the problem when the number of future requests is less than the number of micro-ferries (i.e. if $N < M$). Therefore, we will use the adjusted formulation

$$k_i - k_j + M(x_{ij} + x_{ji} - 1) \leq 0 \quad \forall (i, j) \in \mathcal{K} \quad (27)$$

where the set \mathcal{K} is given by

$$\mathcal{K} = \{(i, j) : \{i, j \in \mathcal{R}, i < j\} \cup \{i, j \in \mathcal{N}, i = j\}\} \quad (28)$$

to allow for loops at the micro-ferry nodes (representing a waiting micro-ferry), which are $\frac{1}{2}R(R-1) - M$ constraints. See Appendix I for more details.

5) *Energy level variables e_i* : With every micro-ferry node $i \in \mathcal{M}$ we associate an initial energy level $e_{0,i}$ by using

$$e_i = e_{0,i} \quad \forall i \in \mathcal{M}. \quad (29)$$

Using the energy-consumption term (9), we can determine the energy levels of the micro-ferries *after* completion of a request j . If a micro-ferry handles request j directly after request i , we have $e_j = e_i - \epsilon_{ij}$ as the energy level after handling request j . Therefore

$$(e_i - e_j - \epsilon_{ij})x_{ij} = 0 \quad \forall i \in \mathcal{R}, j \in \mathcal{N}. \quad (30)$$

These non-linear constraints can be written in an equivalent linear form by choosing an appropriately large constant E , and substituting (30) by the linear inequality constraints

$$e_j - e_i + \epsilon_{ij} + Ex_{ij} \leq E \quad \forall i \in \mathcal{R}, j \in \mathcal{N}, \quad (31a)$$

$$e_i - e_j - \epsilon_{ij} + Ex_{ij} \leq E \quad \forall i \in \mathcal{R}, j \in \mathcal{N}. \quad (31b)$$

Using \underline{e} and \bar{e} to denote the minimum and maximum energy levels of the micro-ferries, the constraints

$$\underline{e} \leq e_i \leq \bar{e}, \quad \forall i \in \mathcal{R} \quad (32)$$

ensure that the schedule is such that micro-ferries will never run out of energy; if there is not enough initial energy in the micro-ferries to conduct all requests, the optimisation problem is infeasible. This could be overcome by including charging in the scheduling, and is considered as future work.

IV. LINEAR PROGRAMMING APPROXIMATION

In the previous section we have described the model for the micro-ferry scheduling problem. The objective function (11) becomes non-linear due to the energy consumption terms ϵ_{ij} defined in (9). We will use an approximation of the speed in order to obtain a mixed-integer linear programming problem.

A. Approximation of the speeds

The speed u_j is related to the variable w_j by (19) as

$$u_j = w_j^{-1}. \quad (33)$$

We approximate this by a piece-wise affine (PWA) function

$$\hat{u}_j = \begin{cases} a_1 w_j + b_1, & w_0 \leq w_j \leq w_1 \\ \vdots & \vdots \\ a_P w_j + b_P, & w_{P-1} \leq w_j \leq w_P \end{cases} \quad (34)$$

where P denotes the amount of sections, $w_0 = \underline{w}$, $w_P = \bar{w}$ as defined in (19), and the scalars w_p are optimisation variables for all $p \in \{1, \dots, P-1\}$. We minimise the error $u_j - \hat{u}_j$ in a least-squares sense to obtain the values of w_1, \dots, w_{P-1} (with $w_p < w_{p+1}$), a_1, \dots, a_P , and b_1, \dots, b_P .

Using the methods described in [13, Section 3.4] we can transform the PWA function (34) into a single function. We introduce $R \cdot P$ binary variables z_{jp} , representing

$$[z_{jp} = 1] \Leftrightarrow [w_j \leq w_p]. \quad (35)$$

The logical rules (35) can be enforced by using

$$w_j - w_p \leq (\bar{w} - \underline{w})(1 - z_{jp}), \quad (36a)$$

$$w_p - w_j \leq (\bar{w} - \underline{w})z_{jp}, \quad (36b)$$

where (36a) ensures $z_{jp} = 0$ when $w_j > w_p$, and (36b) ensures $z_{jp} = 1$ when $w_j < w_p$. Note that all $z_{jq} = 1$ for $q = p+1, \dots, P$ if $z_{jp} = 1$, since $w_p < w_{p+1}$ for all $p = \{1, \dots, P\}$. Using the variables z_{jp} with $z_{jP} = 1$, the PWA function (34) can be written as

$$\hat{u}_j = (A_1 w_j + B_1)z_{j1} + \dots + (A_P w_j + B_P)z_{jP}, \quad (37a)$$

where A_1, \dots, A_P and B_1, \dots, B_P are constants given as

$$A_p = a_p - a_{p+1} \quad \forall p \in \{1, \dots, P-1\}, \quad A_P = a_P, \quad (37b)$$

$$B_p = b_p - b_{p+1} \quad \forall p \in \{1, \dots, P-1\}, \quad B_P = b_P. \quad (37c)$$

Figure 2 shows an example where $2 \leq u_j \leq 5$ using three sections. The continuous, blue curve shows (33), whereas the striped, green lines represent the approximation (37). The dotted, black lines indicate the positions of w_1 and w_2 .

B. Linearised formulation of the energy consumption

Using the approximation (37) of u_j , the approximation of the energy consumption term (12) can be written as

$$\begin{aligned} \hat{\epsilon}_j &= (p_2 \hat{u}_j + p_1 + p_0 \frac{1}{w_j}) \sum_{i=1}^N (c_{ij} + c_{jj}) x_{ij}, \quad (38) \\ &= [p_2 \sum_{p=1}^P \{(A_p w_j + B_p) z_{jp}\} + p_1 + p_0 w_j] \sum_{i=1}^N C_{ij} x_{ij} \end{aligned}$$

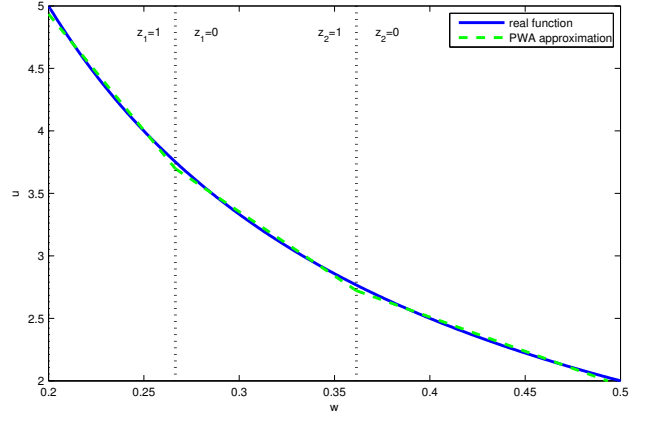


Fig. 2. Real function u and its piece-wise affine approximation \hat{u}

with C_{ij} defined in (10). This equation is non-linear, since it contains multiplications of the variables w_j , z_{jp} and x_{ij} . Using the ideas presented in [13] we will transform (38) into an equivalent linear form as follows.

First we introduce new variables f_j . Notice that by (15a) only one x_{rj} equals one for each j , and hence the sum $\sum_{i=1}^N C_{ij} x_{ij}$ equals the constant C_{rj} for which $x_{rj} = 1$; $C_{ij} x_{ij} = 0$ for all $i \neq r$. If the variable f_j satisfies

$$f_j \leq C_{ij} w_j + (\bar{f} - \underline{f})(1 - x_{ij}) \quad \forall i \in \mathcal{N}, \quad (39a)$$

$$f_j \geq C_{ij} w_j + (\bar{f} - \underline{f})(x_{ij} - 1) \quad \forall i \in \mathcal{N}, \quad (39b)$$

where \underline{f} and \bar{f} are a lower bound and an upper bound on the product $C_{ij} w_j$ respectively, we obtain $f_j \hat{=} w_j \sum_{i=1}^N C_{ij} x_{ij}$; constraints (39a) give upper bounds on f_j with a minimum of $w_j C_{ij}$ when $x_{ij} = 1$; constraints (39b) give lower bounds on f_j , with a maximum of $w_j C_{ij}$ when $x_{ij} = 1$.

Next we define the new variables g_{jp} , and again use the property that only one element $x_{rj} = 1$ for each j . If

$$g_{jp} \leq \bar{g} z_{jp}, \quad g_{jp} \leq \sum_{i=1}^N C_{ij} x_{ij} + \underline{g}(z_{jp} - 1), \quad (40a)$$

$$g_{jp} \geq \underline{g} z_{jp}, \quad g_{jp} \geq \sum_{i=1}^N C_{ij} x_{ij} + \bar{g}(z_{jp} - 1), \quad (40b)$$

where \underline{g} and \bar{g} are a lower bound and an upper bound on the constants C_{ij} respectively, we obtain $g_{jp} \hat{=} z_{jp} \sum_{i=1}^N C_{ij} x_{ij}$.

Finally, we introduce new variables h_{jp} satisfying

$$h_{jp} \leq \bar{f} z_{jp}, \quad h_{jp} \leq f_j + \underline{f}(z_{jp} - 1), \quad (41a)$$

$$h_{jp} \geq \underline{f} z_{jp}, \quad h_{jp} \geq f_j + \bar{f}(z_{jp} - 1), \quad (41b)$$

such that $h_{jp} \hat{=} z_{jp} f_j$. Substituting the non-linear terms in (38) by the variables f_j , g_{jp} , and h_{jp} results in

$$\hat{\epsilon}_j = p_0 f_j + p_1 \sum_{i=1}^N C_{ij} x_{ij} + p_2 \sum_{p=1}^P (A_p h_{jp} + B_p g_{jp}), \quad (42)$$

which is a linear function in the variables f_j , g_{jp} and h_{jp} . The problem becomes a mixed-integer linear program by substituting (11) by the linear approximation

$$\hat{J}_{ec} = \sum_{j=1}^N [p_0 f_j + p_1 \sum_{i=1}^N C_{ij} x_{ij} + p_2 \sum_{p=1}^P (A_p h_{jp} + B_p g_{jp})]. \quad (43)$$

V. CONCLUSIONS AND FUTURE WORK

In this paper a method is proposed to schedule the transportation requests of micro-ferries. This method can be seen as a novel variant of the travelling salesman problem, or more specifically the vehicle routing problem, where we include variable speeds and energy consumption. Besides influencing the energy consumption, the variable speeds add flexibility in the time it takes to handle a request, making it possible to satisfy time window constraints that otherwise would not be feasible. The exact formulation results in a mixed-integer non-linear programming problem, but by using a piece-wise affine approximation of the speed we can present the problem as a mixed-integer linear programming problem.

This paper presents the first modelling results of the micro-ferry scheduling problem, and serves as a basis for future work. The current work can be expanded to take into account the charging of the micro-ferries, to make it possible to obtain schedules for long time-periods. Using e.g. a rolling horizon approach, one can then use the knowledge about the daily transportation needs by including expected requests in the problem to account for trends in the transportations. The linearisation provides accurate results, but by introducing new binary variables the mixed-integer linear program becomes hard to solve for large numbers of requests. Therefore, alternative methods to solve the original (non-linear) problem might be developed.

APPENDIX I NODE CURRENTS WITH LOOPS

This appendix provides a more detailed derivation of the inequality constraints (27) and their associated set \mathcal{K} defined in (28). Recall that the set \mathcal{R} is the concatenation of the sets \mathcal{M} (associated with the current requests/micro-ferries) and \mathcal{N} (associated with the future requests); node index i satisfies $i \in \{1, \dots, M\}$ if $i \in \mathcal{M}$ and $i \in \{M+1, \dots, R\}$ if $i \in \mathcal{N}$. The constraints of (26) can be split into four sets:

- i) arcs (i, j) from micro-ferry nodes ($i \in \mathcal{M}$) towards micro-ferry nodes ($j \in \mathcal{M}$);
- ii) arcs (i, j) from micro-ferry nodes ($i \in \mathcal{M}$) towards future request nodes ($j \in \mathcal{N}$);
- iii) arcs (i, j) from future request nodes ($i \in \mathcal{N}$) towards micro-ferry nodes ($j \in \mathcal{M}$);
- iv) arcs (i, j) from future request nodes ($i \in \mathcal{N}$) towards future request nodes ($j \in \mathcal{N}$).

If we would use the inequality constraints (26), there will be R constraints where $i = j$ such that

$$k_i - k_i + (x_{ii} + x_{ii} - 1) = 2x_{ii} - 1 \leq 0, \quad (44)$$

and hence —since $x_{ii} \in \{0, 1\}$ — this can only be satisfied if $x_{ii} = 0$. Within the network setting this means that a node i cannot have a loop to itself; this result is desired for the nodes associated with the future request in \mathcal{N} , but not for the nodes associated with the current requests \mathcal{M} .

Within the network of requests $\mathcal{G} = (\mathcal{R}, \mathcal{A})$, an arc (i, j) with $j \in \mathcal{M}$ represents the final arc in the tour for micro-ferry $k_j \in \mathcal{M}$; that is, request i is the last request handled

by micro-ferry k_j . Therefore, a loop at a node $j \in \mathcal{M}$ would mean that vehicle k_j will handle its current request, and does not take on any future requests. This behaviour is *desirable*, as it can be better (with respect to the objectives) to not use a particular micro-ferry; when the number of future requests is smaller than the number of micro-ferries it is *necessary*, since there are not enough nodes belonging to set \mathcal{N} to create M tours. Therefore, we should allow the possibility that $x_{jj} = 1$ for $j \in \mathcal{M}$. This is done by substituting (26) with

$$k_i - k_j + M(x_{ij} + x_{ji}) \leq M \quad \forall i \in \mathcal{M}, j \in \mathcal{M}, i < j \quad (45a)$$

$$k_i - k_j + M(x_{ij} + x_{ji}) \leq M \quad \forall i \in \mathcal{M}, j \in \mathcal{N}, i \leq j \quad (45b)$$

$$k_i - k_j + M(x_{ij} + x_{ji}) \leq M \quad \forall i \in \mathcal{N}, j \in \mathcal{M}, i \leq j \quad (45c)$$

$$k_i - k_j + M(x_{ij} + x_{ji}) \leq M \quad \forall i \in \mathcal{N}, j \in \mathcal{N}, i \leq j \quad (45d)$$

where (45a) excludes the set of constraints where $i = j$ for $i, j \in \mathcal{M}$, such that loops are allowed for these nodes. Note that (45c) does not add any constraints, since $i > j$ if $i \in \mathcal{N}$ and $j \in \mathcal{M}$; furthermore, in (45b) we can use $i < j$ since $i < j$ if $i \in \mathcal{M}$ and $j \in \mathcal{N}$. Inequality constraints (45) can thus be described as a set of constraints where $i < j$ for $i, j \in \mathcal{R}$ plus a set of constraints where $i = j$ for $i, j \in \mathcal{N}$. We obtain

$$k_i - k_j + M(x_{ij} + x_{ji} - 1) \leq 0 \quad \forall i, j \in \mathcal{R}, i < j \quad (46a)$$

$$k_i - k_j + M(x_{ij} + x_{ji} - 1) \leq 0 \quad \forall i, j \in \mathcal{N}, i = j \quad (46b)$$

where (46a) ensures that $k_i = k_j$ when $\theta_{ij} = 1$ or $\theta_{ji} = 1$, while (46b) prevents the existence of loops for the nodes associated with the future requests. This results in the constraints (27) for the set \mathcal{K} of arcs (i, j) given by (28).

REFERENCES

- [1] Stadshavens Rotterdam, “1600 ha - implementation programme 2007–2015.” [Online]. Available: stadshavensrotterdam.nl
- [2] M. Savelsbergh and M. Sol, “The general pickup and delivery problem,” *Transportation Science*, vol. 29, no. 1, pp. 17–29, Feb. 1995.
- [3] S. Ropke and J. Cordeau, “Branch and cut and price for the pickup and delivery problem with time windows,” *Transportation Science*, vol. 43, pp. 267–286, 2009.
- [4] T. Bektas, “The multiple traveling salesman problem: an overview of formulations and solution procedures,” *Omega*, vol. 34, no. 3, pp. 209–219, 2006.
- [5] R. V. Kulkarni and P. R. Bhave, “Integer programming formulations of vehicle routing problems,” *European Journal of Operational Research*, vol. 20, no. 1, pp. 58–67, 1985.
- [6] J. Cordeau and G. Laporte, “The dial-a-ride problem (DARP): Variants, modeling issues and algorithms,” *4OR: A Quarterly Journal of Operations Research*, vol. 1, no. 2, pp. 89–101, 2003.
- [7] S. Itani, E. Frazzoli, and M. Dahleh, “Dynamic traveling repairperson problem for dynamic systems,” in *Proc. of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, Dec. 2008, pp. 465–470.
- [8] T. Fossen, *Marine Control Systems - Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*, 1st ed. Marine Cybernetics, 2002.
- [9] G. Laporte, “The traveling salesman problem: an overview of exact and approximate algorithms,” *European Journal of Operational Research*, vol. 59, no. 2, pp. 231–247, 1992.
- [10] H. Taha, *Operations Research: An Introduction*, 4th ed. New York: Macmillan Publishing Company, 1987.
- [11] C. Miller, A. Tucker, and R. Zemlin, “Integer programming formulation of traveling salesman problems,” *Journal of Association for Computing Machinery*, vol. 7, no. 4, pp. 326–329, 1960.
- [12] M. Desrochers and G. Laporte, “Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints,” *Operations Research Letters*, vol. 10, no. 1, pp. 27–36, 1991.
- [13] A. Bemporad and M. Morari, “Control of systems integrating logic, dynamics, and constraints,” *Automatica*, vol. 35, pp. 407–427, 1999.