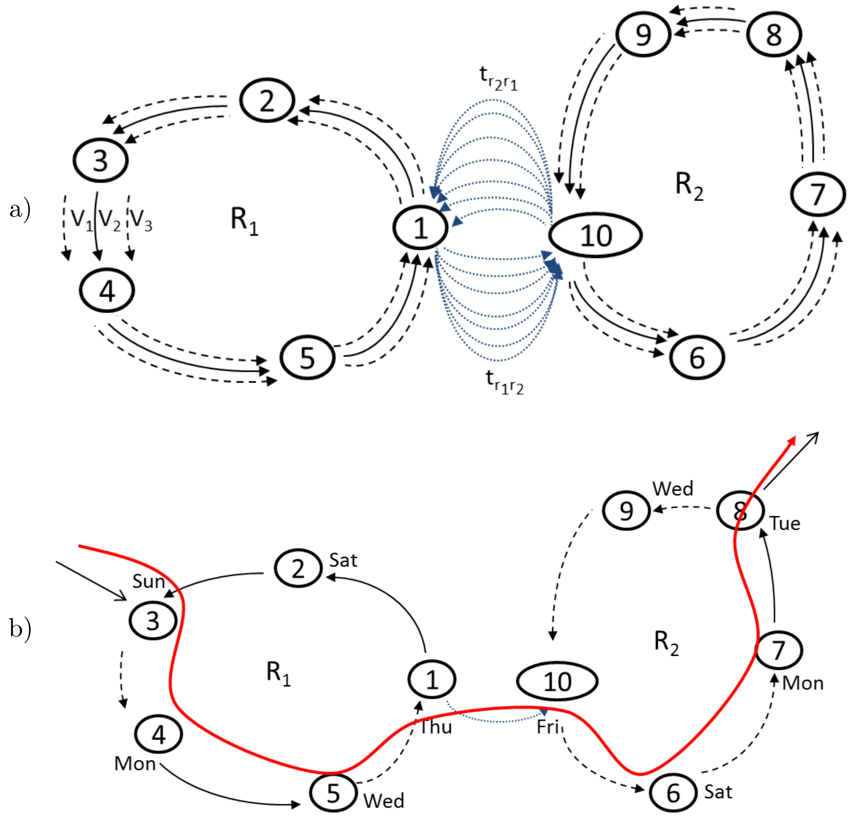

Figure 1: Figure a) a transportation network used for speed and schedule optimization (the problem considered in the BMP). Figure b) the flow through a network at a spcific speed and schedule (the problem considered in the PBSP). The nodes correspond to port calls, which are numbered consecutively around each rotation. The ten port calls correspond to nine physical ports (the port calls 1 and 10 correspond to the same physical port) and it is possible to transship between rotation R_1 and R_2 by using a transshipment arc from 1 to 10.

A Optimal Time Scheduling

It is possible to add time scheduling constraints to the model to include determination of an optimal time schedule while satisfying the transit time restrictions. The time schedule determines the timing of events, such as port calls, along a rotation. For each port we introduce a variable to determine the departure time, and at the same time we introduce variables to reflect transshipment time between rotations. These are coupled with the flow variables such that it is possible to consider the influence of schedules in the flow calculations. The time scheduling and transshipment decisions are included in the BMP. Duplicates of the transshipment arcs are considered with different layover time corresponding to all possible schedules, and a binary variable is associated with each of the transshipment arcs. The selected transshipment arc is included in the PBSP. If the departure time for some or all ports are given a priori, this is easy to include by fixing part of the network.

Figure 1 is an extension of the example in Figure 3 and Figure 1 a) shows the transportation network before the schedule and speeds have been selected. The transshipment time from rotation R_1 to R_2 will depend on the schedule of the two rotations, but since both schedule and speed is variable we need to consider different transshipment arcs (blue dotted lines in the figure). In this case there are seven arcs corresponding to a transshipment time of one to seven days. Figure b) shows an example flow from 3 to 8 where the schedule and speeds have been selected. Rotation R_1 calls port 1 every Thursday and R_2 departs from the same port (represented by a different port call) every Friday, so the transshipment arc used in this case is the one corresponding to one day. This way it is possible to calculate the transit time from 3 to 8 as the length of each sailing leg divided by the selected speed for each of the legs plus the transshipment time corresponding to the selected schedule. If the total transit time from the physical port corresponding to 3 to the physical port corresponding to 8 is longer than what is allowed for a commodity going on this path, there is no feasible path and it may be worth adjusting the speeds and the schedule.

A.1 Modeling

We wish to determine when each port is visited by which rotation and how this influences the achievable transshipment times and thereby the cargo routing. For each rotation the schedule (i.e. arrival and departure times) is determined in all ports for all rotations. The time it takes to transship between two different rotations visiting the same port is determined by the arrival in the port for each of the rotations and some buffer time. In the following we assume for simplicity weekly rotations, but the model can be extended to accommodate bi-weekly frequencies. The set of all port calls is I and in the example of nine physical ports in Figure 1 this corresponds to the port calls 1 to 10 which are consecutively numbered along each rotation. For each rotation $r \in R$ we assign a starting port call σ_r a priori (port call 1 and 6 in Figure 1) and use this as reference for the schedule of the subsequent ports in each of the rotations. The starting port will always be the call with the lowest index within the rotation. Each rotation consist of several port calls, and in the case of butterfly rotations the same port may have multiple corresponding port calls. The set of physical ports is Q and the set of port calls in port $q \in Q$ is $I(q)$. The arrival time at port call $i \in I$ given in days is determined by the continuous decision variable $T_i \in \mathbb{R}$ and the integer decision variable $w_{i'i''} \in \mathbb{Z}$ is the offset in weeks between two port calls for the same port, i' and i'', i.e., $w_{i'i''}$ is not necessarily equal to $w_{i''i'}$. Usually two port calls i' and i'' at port p correspond to two different rotations, but for butterfly rotations they can correspond to two calls from the same rotation. Additionally, $g_{i'i''}$ is the necessary transshipment buffer time between arrival of port call i' and departure of port call i'' in port p. The constant $\bar{t}_i$ specify the length of the stay of port call i and the continuous decision variable $\hat{T}_{i'i''} \in \mathbb{R}$ is the transshipment time from port call i' to port call i'' in a specific port. The set of multiarcs between port call i and $i+1$ is $A(i)$, i.e., arcs at different speeds between two consecutive ports on the same rotation. All the ordered pairs of rotations visiting port q are denoted $(i', i'') \in Q^2(q)$. For each port q with a transshipment opportunity and for each $(i', i'') \in Q^2(q)$, the set of arcs available for transshipment is given by the set $A(i', i'')$. The capacity of the corresponding arc, $u_{i'i''}$, is given by the minimum capacity of the rotation corresponding to port call i' and the rotation corresponding to i'' in a given port. All transshipment arcs have an associated transshipment time, t_a, and we include binary variables, x_a for $a \in A(i', i'')$ that selects whether a transshipment arc is used. The time scheduling part of the model is

$$T_i = T_{i-1} + \sum_{a \in A(i-1)} t_a x_a \qquad i \in I \setminus \cup_{r \in R} \{\sigma_r\} \tag{1}$$

$$\hat{T}_{i'i''} = (T_{i''} + \bar{t}_{i''}) - T_{i'} + 7 w_{i'i''} \qquad q \in Q,\ (i', i'') \in Q^2(q) \tag{2}$$

$$\sum_{a \in A(i'i'')} t_a x_a \geq \hat{T}_{i'i''} \qquad q \in Q,\ (i', i'') \in Q^2(q) \tag{3}$$

$$\sum_{a \in A(i'i'')} x_a = 1 \qquad q \in Q,\ (i', i'') \in Q^2(q) \tag{4}$$

$$g_{i'i''} \leq \hat{T}_{i'i''} \leq g_{i'i''} + 7 \qquad q \in Q,\ (i', i'') \in Q^2(q) \tag{5}$$

$$T_i \in \mathbb{R}_+ \qquad i \in I \tag{6}$$

$$\hat{T}_{i'i''} \in \mathbb{R}_+ \qquad q \in Q,\ (i', i'') \in Q^2(q) \tag{7}$$

$$w_{i''i'} \in \mathbb{Z} \qquad q \in Q,\ (i', i'') \in Q^2(q) \tag{8}$$

$$x_a \in \{0, 1\} \qquad q \in Q,\ (i', i'') \in Q^2(q),\ a \in A(i', i'') \tag{9}$$

The relation of the departure time for two consecutive ports is given by Constraints (1). Notice that we in Constraints (1) let i run in the elements of I except the starting port call of each rotation. The reason is that we need to avoid a cyclic definition of T_i, which would be infeasible. The transshipment time in port p from port call i' to port call i'' is determined by Constraints (2) and Constraints (3) and (4) makes sure we only select one transshipment arc and that it is feasible. Constraints (5) limits the possible transshipment time. If we have a schedule determined by the hour there are going to be 168 transshipment arcs for each feasible (i', i'')-combination, but we can reduce the number of available transshipment arcs such that we overestimate the transshipment time. There can e.g. be one available arc for each day, i.e., 7 arcs, and for some ports we can have higher accuracy than others by including more arcs.

To illustrate the offset variable consider an instance with hourly accuracy where $\bar{t}_{i''} = 8$ and $g_{i'i''} = 10$ then if $T_{i'} = 24$ and $T_{i''} = 48$ we get that $w_{i'i''} = 0$ and $\hat{T}_{i'i''} = 48 + 8 - 24 = 32$. If there is a too tight schedule, i.e., $T_{i'} = 24$ and $T_{i''} = 24$, we get that the commodity will have to wait because of the buffer time and we get that $w_{i'i''} = 1$ and $\hat{T}_{i'i''} = 24 + 8 - 24 + 168 = 176$. To illustrate the influence of the schedule on longer rotations consider first $T_{i'} = 13 * 24 = 312$ and $T_{i''} = 24$, then we get $w_{i'i''} = 2$ and $\hat{T}_{i'i''} = 24 + 8 - 312 + 2 * 168 = 56$. Conversely if $T_{i'} = 24$ and $T_{i''} = 312$, then we get $w_{i'i''} = -1$ and $\hat{T}_{i'i''} = 312 + 8 - 24 - 168 = 128$.

A.2 Including the Time Scheduling Part in the Benders Decomposition

Similarly to the coupling constraints for the sailing arcs, we can introduce coupling constraints for the transshipment arcs

$$\sum_{k \in K} \sum_{p \in P(a,k)} y_p^k \leq x_a u_{i'i''} \qquad a \in A(i', i''),\ q \in Q,\ (i', i'') \in Q^2(q) \tag{10}$$

Then including the time scheduling part in the Benders decomposition lead to a slightly modified sub-problem where we are now considering the transshipment arcs as well such that the PBSP is given by

$$\min \sum_{k \in K} \sum_{p \in P^k} r_p^k y_p^k \tag{11}$$

subject to

$$\sum_{p \in P^k} y_p^k \leq d^k \qquad k \in K \tag{12}$$

$$\sum_{k \in K} \sum_{p \in P(a,k)} y_p^k \leq \bar{x}_a q_a \qquad q \in Q,\ i \in I(q),\ a \in A(i) \tag{13}$$

$$\sum_{k \in K} \sum_{p \in P(a,k)} y_p^k \leq \bar{x}_a u_{i'i''} \qquad q \in Q,\ (i', i'') \in Q^2(q),\ a \in A(i', i'') \tag{14}$$

$$y_p^k \in \mathbb{R}_+ \qquad k \in K, p \in P^k \tag{15}$$

This can still be solved using column generation, but now the column generation sub-problem also contains dual variables corresponding to transshipment arcs. The BMP will be (20)-(26) + (1)-(9).

B Generating Additional Benders Cuts

It is possible to obtain additional Benders cuts in each iteration of the algorithm based on a solution to the multi-commodity flow problem. When the sub-problem has been solved we modify the problem to have an additional constraint restricting the objective to be at least as good as the optimal solution but maximizing some distance to the solution, e.g.:

$$\max \sum_{k \in K} \sum_{p \in P^k} |\bar{y}_p^k - y_p^k| \tag{16}$$

subject to

$$\sum_{p \in P^k} y_p^k \leq d^k \qquad k \in K \tag{17}$$

$$\sum_{k \in K} \sum_{p \in P(a,k)} y_p^k \leq q_a x_a \qquad (a) \in A \tag{18}$$

$$\sum_{k \in k} \sum_{p \in P^k} r_p^k y_p^k \leq \sum_{k \in K} \sum_{p \in P^k} r_p^k \bar{y}_p^k \tag{19}$$

$$y_p^k \in \mathbb{R}_+ \qquad k \in K,\ p \in P^k \tag{20}$$

where $\bar{y}_p^k$ is an optimal solution and we use the set of already generated paths to find an alternative solution. However, the objective (16) is non-linear so we modify it such that if $\bar{y}_p^k = 0$ then we take $(y_p^k - \bar{y}_p^k)$ and if $\bar{y}_p^k = d^k$ then we take $(\bar{y}_p^k - y_p^k)$. If $0 < \bar{y}_p^k < d^k$ we use $(\bar{y}_p^k - y_p^k)$ as the objective with probability $\frac{\bar{y}_p^k}{d^k}$ and $y_p^k - \bar{y}_p^k$ otherwise. Leaving out the constant term, the objective becomes

$$\max \sum_{k \in K} \left(\sum_{p \in P^k : \bar{y}_p^k = 0} y_p^k - \Big(\sum_{p \in P^k : \bar{y}_p^k = d^k} y_p^k \Big) + \sum_{p \in P^k : 0 < \bar{y}_p^k < d^k} X_p y_p^k \right) \tag{21}$$

Where X_p is a random variable of ±1 with probability $\frac{\bar{y}_p^k}{d^k}$. Using only a reduced set of columns when solving the primal problem will lead to the optimal primal solution, but the corresponding dual solution may not be optimal/feasible as only a subset of constraints are considered. To obtain appropriate dual values, the objective is changed to the original objective, such that constraint (16) is removed and the problem is resolved using the solution as a warm start. The new dual solution is checked by solving the pricing problem in multi-commodity flow problem once, and if no reduced cost columns are returned an additional cut is added based on the new dual variables. If a reduced cost column is found, we do not add a cut.

Similarly we could solve the multi-commodity flow problem with the set of columns already obtained with an interior point method to obtain an alternative Benders cut based on a solution centered towards the interior. Again we need to check the multi-commodity flow pricing problem and only add the cut if no reduced cost paths are found.

C Computational Results

Traditional implementation	Branch-and-cut implementation	LP-relaxation cuts	Warm start cut	Strengthened cuts	Node relaxation cuts	Valid inequalities	Improvement in profit (%)	Final gap (FUB-FLB)/|FUB| (%)	Final UB gap (FUB*-FUB)/|FUB*| (%)	Final LB gap (FUB*-FLB)/|FUB*| (%)	Initial LB gap (FUB*-ILB)/|FUB*| (%)	Runtime (seconds)	Solved	Improvement in profit (%)	Final gap (FUB-FLB)/|FUB| (%)	Final UB gap (FUB*-FUB)/|FUB*| (%)	Final LB gap (FUB*-FLB)/|FUB*| (%)	Initial LB gap (FUB*-ILB)/|FUB*| (%)	Runtime (seconds)	Solved
Setting							**Fixed deployment**							**Flexible deployment**						
F450																				
×							0.0	0.0	0.0	0.0	-	8	1	0.0	0.0	0.0	0.0	-	5	1
	×						0.0	0.0	0.0	0.0	0.0	4	1	0.0	0.0	0.0	0.0	0.0	5	1
×		×					0.0	0.0	0.0	0.0	0.0	4	1	0.0	0.0	0.0	0.0	0.1	8	1
	×	×					0.0	0.0	0.0	0.0	0.0	5	1	0.0	0.0	0.0	0.0	0.0	8	1
×		×	×				0.0	0.0	0.0	0.0	0.0	8	1	0.0	0.0	0.0	0.0	0.1	8	1
	×	×	×				0.0	0.0	0.0	0.0	0.0	4	1	0.0	0.0	0.0	0.0	0.0	8	1
×		×	×	×			0.0	0.0	0.0	0.0	0.0	4	1	0.0	0.0	0.0	0.0	0.1	9	1
	×	×	×	×			0.0	0.0	0.0	0.0	0.0	5	1	0.0	0.0	0.0	0.0	0.0	10	1
×		×	×			×	0.0	0.0	0.0	0.0	0.0	8	1	0.0	0.0	0.0	0.0	0.0	4	1
	×	×	×			×	0.0	0.0	0.0	0.0	0.0	10	1	0.0	0.0	0.0	0.0	0.0	11	1
	×	×	×		×		0.0	0.0	0.0	0.0	0.0	5	1	0.0	0.0	0.0	0.0	0.0	8	1
F800																				
×							0.2	0.0	0.0	0.1	-	33	1	0.2	0.0	0.0	0.1	-	41	1
	×						0.0	0.3	0.2	0.1	0.1	9	1	0.0	0.3	0.2	0.1	0.1	18	1
×		×					0.0	0.0	0.2	0.2	0.6	21	1	0.0	0.0	0.2	0.2	0.6	47	1
	×	×					0.0	0.5	0.3	0.1	0.1	33	1	0.0	0.6	0.4	0.1	0.1	64	1
×		×	×				0.0	0.0	0.2	0.2	0.6	35	1	0.0	0.0	0.2	0.2	0.6	36	1
	×	×	×				0.0	0.3	0.2	0.1	0.1	31	1	0.0	0.3	0.2	0.1	0.1	71	1
×		×	×	×			0.0	0.0	0.2	0.2	0.6	41	1	0.0	0.0	0.2	0.2	0.6	45	1
	×	×	×	×			0.0	0.3	0.2	0.1	0.1	44	1	0.0	0.3	0.2	0.1	0.1	98	1
×		×	×			×	0.0	0.0	0.2	0.3	0.2	78	1	0.0	0.0	0.2	0.3	1.7	72	1
	×	×	×			×	0.0	0.3	0.2	0.1	0.1	148	1	0.0	0.3	0.2	0.1	0.1	178	1
	×	×	×		×		0.0	0.3	0.2	0.1	0.1	33	1	0.0	0.3	0.2	0.1	0.1	71	1
P1200																				
×							0.5	0.7	0.0	0.7	-	26	1	6.8	13.1	1.4	11.5	-	10800	0
	×						0.5	0.6	0.0	0.5	0.6	10	1	8.0	17.9	0.2	17.6	69.4	10800	0
×		×					0.1	0.0	0.4	0.4	0.5	23	1	6.3	10.4	1.9	8.3	10.8	10800	0
	×	×					0.1	0.7	0.4	0.2	0.2	27	1	7.2	7.4	1.0	6.3	6.4	10800	0
×		×	×				0.1	0.4	0.4	0.0	1.0	31	1	6.7	6.5	1.5	4.9	6.6	10800	0
	×	×	×				0.0	0.8	0.5	0.2	0.2	21	1	8.3	6.3	0.0	6.3	6.3	10800	0
×		×	×	×			0.1	0.4	0.4	0.0	1.0	30	1	5.1	8.1	3.0	4.8	6.6	10800	0
	×	×	×	×			0.0	0.8	0.5	0.2	0.2	31	1	6.9	7.9	1.3	6.5	6.5	10800	0
×		×	×			×	0.5	0.0	0.1	0.1	1.0	31	1	6.2	24.0	1.9	21.6	23.8	10800	0
	×	×	×			×	0.0	0.8	0.5	0.2	0.2	35	1	7.9	6.9	0.4	6.5	6.6	10800	0
	×	×	×		×		0.0	0.8	0.5	0.2	0.2	27	1	5.8	8.8	2.3	6.3	6.4	10800	0

Table 1: Computational results for individual vessel classes in WorldSmall.

Traditional implementation	Branch-and-cut implementation	LP-relaxation cuts	Warm start cut	Strengthened cuts	Node relaxation cuts	Valid inequalities	Improvement in profit (%)	Final gap (FUB-FLB)/|FUB| (%)	Final UB gap (FUB*-FUB)/|FUB*| (%)	Final LB gap (FUB*-FLB)/|FUB*| (%)	Initial LB gap (FUB*-ILB)/|FUB*| (%)	Runtime (seconds)	Solved	Improvement in profit (%)	Final gap (FUB-FLB)/|FUB| (%)	Final UB gap (FUB*-FUB)/|FUB*| (%)	Final LB gap (FUB*-FLB)/|FUB*| (%)	Initial LB gap (FUB*-ILB)/|FUB*| (%)	Runtime (seconds)	Solved
Setting							**Fixed deployment**							**Flexible deployment**						
P2400																				
×							0.3	0.5	0.2	0.3	-	42	1	9.9	75.3	2.6	70.8	-	10800	0
	×						0.5	0.8	0.0	0.8	0.9	8	1	9.3	136.2	3.2	128.7	139.3	10800	0
×		×					0.3	0.3	0.2	0.1	1.0	33	1	6.6	54.9	17.2	28.2	33.5	10800	0
	×	×					0.5	0.8	0.0	0.7	0.7	18	1	11.7	27.8	1.0	26.6	27.4	10800	0
×		×	×				0.0	0.9	0.5	0.4	1.4	38	1	7.8	39.0	4.5	32.8	37.0	10800	0
	×	×	×				0.5	0.8	0.0	0.7	0.7	19	1	9.5	30.7	2.9	26.9	27.6	10800	0
×		×	×	×			0.0	0.9	0.5	0.4	1.4	27	1	0.0	49.6	11.4	32.6	36.8	10800	0
	×	×	×	×			0.5	0.8	0.0	0.7	0.7	31	1	7.7	33.0	4.5	26.9	27.9	10800	0
×		×	×			×	0.0	1.0	0.5	0.4	1.4	8	1	0.0	93.8	11.4	71.8	73.7	10800	0
	×	×	×			×	0.5	0.8	0.5	0.7	0.7	114	1	8.2	30.9	4.1	25.5	25.7	10800	0
	×	×	×		×		0.5	0.8	0.0	0.7	0.7	19	1	7.3	32.6	4.9	26.1	27.6	10800	0
PostP																				
×							0.0	19.1	0.9	17.9	-	10800	0	7.1	84.1	2.4	79.6	-	10800	0
	×						0.3	13.5	0.5	12.9	126.0	10800	0	2.4	151.4	6.7	134.7	137.4	10800	0
×		×					0.2	1.0	0.6	0.3	1.4	1931	1	2.0	103.4	7.1	89.0	92.9	10800	0
	×	×					0.6	1.2	0.2	1.0	1.3	10800	0	1.3	30.4	7.7	20.4	21.6	10800	0
×		×	×				0.4	1.0	0.5	0.5	1.7	7068	1	1.7	62.7	7.3	50.8	55.6	10800	0
	×	×	×				0.7	1.0	0.1	0.9	1.2	2639	1	0.0	32.0	8.9	20.3	21.6	10800	0
×		×	×	×			0.3	1.1	0.5	0.5	1.7	10800	0	3.0	60.8	6.1	50.9	56.0	10800	0
	×	×	×	×			0.8	0.9	0.0	0.9	1.2	3770	1	3.6	27.4	5.5	20.3	21.6	10800	0
×		×	×			×	0.0	11.1	0.8	10.2	11.5	10800	0	0.0	146.0	8.9	124.2	127.9	10800	0
	×	×	×			×	0.8	1.0	0.8	1.0	1.1	1369	1	9.7	19.1	0.0	19.1	19.8	10800	0
	×	×	×		×		0.8	1.0	0.0	1.0	1.2	2465	1	0.0	32.1	8.9	20.3	21.6	10800	0
SuperP																				
×							0.0	0.7	0.3	0.4	-	8	1	0.0	0.8	0.2	0.6	-	22	1
	×						0.0	0.3	0.0	0.3	0.3	6	1	0.0	0.8	0.2	0.6	2.1	14	1
×		×					0.0	0.2	0.1	0.1	2.2	8	1	0.0	0.8	0.2	0.6	2.6	27	1
	×	×					0.0	0.2	0.0	0.2	0.2	18	1	0.0	0.5	0.2	0.3	1.7	69	1
×		×	×				0.0	0.1	0.0	0.1	1.0	6	1	0.0	0.3	0.0	0.3	1.9	25	1
	×	×	×				0.0	0.2	0.0	0.2	0.2	17	1	0.0	0.8	0.0	0.8	1.7	48	1
×		×	×	×			0.0	0.1	0.0	0.1	1.0	7	1	0.0	0.3	0.0	0.3	1.9	29	1
	×	×	×	×			0.0	0.2	0.0	0.2	0.2	26	1	0.0	0.8	0.0	0.8	1.7	129	1
×		×	×			×	0.0	0.1	0.0	0.1	5.2	7	1	0.0	0.0	0.0	1.2	0.7	17	1
	×	×	×			×	0.0	0.2	0.0	0.2	0.2	27	1	0.0	0.2	0.0	0.1	0.1	25	1
	×	×	×		×		0.0	0.2	0.0	0.2	0.2	17	1	0.0	0.8	0.0	0.7	1.7	79	1

Table 2: Computational results for individual vessel classes in WorldSmall.