Gabriel André Homsi

Ship routing and speed optimization with heterogeneous fuel consumption profiles

DISSERTAÇÃO DE MESTRADO

DEPARTAMENTO DE INFORMÁTICA
Programa de Pós-Graduação em Informática

Rio de Janeiro
February 2018

Pontifícia Universidade Católica
do Rio de Janeiro

Gabriel André Homsi

Ship routing and speed optimization with heterogeneous fuel consumption profiles

Dissertação de Mestrado

Dissertation presented to the Programa de Pós-Graduação em Informática of the Departamento de Informática, PUC-Rio as partial fulfillment of the requirements for the degree of Mestre em Informática.

Advisor: Prof. Thibaut Victor Gaston Vidal

Rio de Janeiro
February 2018

Pontifícia Universidade Católica
do Rio de Janeiro

Gabriel André Homsi

Ship routing and speed optimization with heterogeneous fuel consumption profiles

Dissertation presented to the Programa de Pós-Graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Informática. Approved by the Undersigned Examination Committee.

Prof. Thibaut Victor Gaston Vidal
Advisor
Departamento de Informática — PUC-Rio

Prof. Rafael Martinelli Pinto
Departamento de Engenharia Industrial — PUC-Rio

Prof. Eduardo Uchoa Barboza
Departamento de Engenharia de Produção — UFF

Prof. Márcio da Silveira Carvalho
Vice Dean of Graduate Studies
Centro Técnico Científico — PUC-Rio

Rio de Janeiro, February 8^{th}, 2018

Gabriel André Homsi

Bachelor's in Computer Science at the Rio de Janeiro State University (2016).

Bibliographic data

Homsi, Gabriel André

Ship routing and speed optimization with heterogeneous fuel consumption profiles / Gabriel André Homsi ; advisor: Thibaut Victor Gaston Vidal. — 2018.
67 f. : il. ; 30 cm

Dissertação (Mestrado em Informática)-Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2018.
Inclui bibliografia

1. Informática – Teses. 2. Transportes;. 3. Roteamento de navios;. 4. Meta-heurísticas;. 5. Branch-and-price;. 6. Otimização de velocidade;. 7. Otimização convexa;. I. Vidal, Thibaut Victor Gaston. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD: 004

Acknowledgments

I devote this space to acknowledge the big influence that the following teachers had on my computer science career: Thibaut Vidal, Rafael Martinelli, Ricardo Costa, and Daniel Fleischman.

This research project was made possible thanks to funds provided by CNPq and CAPES.

Abstract

Homsi, Gabriel André; Vidal, Thibaut Victor Gaston (advisor). **Ship routing and speed optimization with heterogeneous fuel consumption profiles**. Rio de Janeiro, 2018. 67p. Dissertação de Mestrado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

The shipping industry is essential for international trade. However, in the wake of the 2008 financial crisis, this industry was severely hit. In these times, transportation companies can only obtain profit if their fleet is routed effectively. In this work, we study a class of ship routing problems related to the Pickup and Delivery Problem with Time Windows. To solve these problems, we introduce a heuristic and an exact method. The heuristic method is a hybrid metaheuristic with a set-partitioning-based large neighborhood, while the exact method is a branch-and-price algorithm. We conduct experiments on a benchmark suite based on real-life shipping segments. The results obtained show that our algorithms largely outperform the state-of-the-art methodologies. Next, we adapt the benchmark suite to model a ship routing problem where the speed on each sailing leg is a decision variable, and fuel consumption per time unit is a convex function of the ship speed and payload. To solve this new ship routing problem with speed optimization, we extend our metaheuristic to find optimal speed decisions on every local search move evaluation. Our computational experiments demonstrate that such approach can be highly profitable, with only a moderate increase in computational effort.

Keywords

Transportation; Ship routing; Metaheuristics; Branch-and-price; Speed optimization; Convex optimization;

Resumo

Homsi, Gabriel André; Vidal, Thibaut Victor Gaston. **Roteamento de navios e otimização de velocidade com perfis de consumo de combustível heterogêneos**. Rio de Janeiro, 2018. 67p. Dissertação de Mestrado — Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A indústria de transporte marítimo é essencial para o comércio internacional. No entanto, no despertar da crise financeira de 2008, essa indústria foi severamente atingida. Nessas ocasiões, empresas de transporte só são capazes de obter lucro se suas frotas forem roteadas de forma eficaz. Neste trabalho, nós estudamos uma classe de problemas de roteamento de navios relacionados ao *Pickup and Delivery Problem with Time Windows*. Para resolver esses problemas, nós introduzimos um método heurístico e um exato. O método heurístico é uma meta-heurística híbrida com uma vizinhança larga baseada em *set partitioning*, enquanto o método exato é um algoritmo de *branch-and-price*. Nós conduzimos experimentos em um conjunto de instâncias baseadas em rotas de navios reais. Os resultados obtidos mostram que nossos algoritmos superam as metodologias estado da arte. Em seguida, nós adaptamos o conjunto de instâncias para modelar um problema de roteamento de navios no qual a velocidade em cada segmento de rota é uma variável de decisão, e o consumo de combustível por unidade de tempo é uma função convexa da velocidade e carga do navio. A fim de resolver esse novo problema de roteamento de navios com otimização de velocidade, nós estendemos nossa meta-heurística para encontrar decisões de velocidade ótimas em toda avaliação de solução vizinha de uma busca local. Nossos experimentos demonstram que essa abordagem pode ser altamente rentável, e que requer apenas um aumento moderado de recursos computacionais.

Palavras-chave

Transportes; Roteamento de navios; Meta-heurísticas; Branch-and-price; Otimização de velocidade; Otimização convexa;

Contents

List of Figures

List of Tables

1
Introduction

International trade depends heavily on ship transportation, as it is the only cost-effective mode of transportation of large volumes over long distances. It is common to distinguish between three main modes of operation in maritime transportation: liner, industrial, and tramp shipping. Liner shipping, which includes container shipping, has similarities to a bus service: fixed schedules and itineraries must be followed. In industrial shipping, the operator owns the cargoes and controls the fleet, trying to minimize the cost of transporting all its cargoes. Finally, a tramp shipping operator follows the availability of cargoes in the market, often transporting a mix of mandatory and optional cargoes with the goal of maximizing profit.

In this work, we focus on heuristic and exact methodologies for a class of Industrial and Tramp Ship Routing and Scheduling Problems (ITSRSPs), along with problem extensions where the sailing speed of ships is a decision variable and the fuel consumption is a convex function of speed and payload. In this class of problems, a shipping company has a mix of mandatory and optional cargoes for transportation. Each cargo in the given planning period must be picked up at its given loading port within a specified time window, transported, and then delivered at its corresponding unloading port, also within a given time window. The shipping company controls a heterogeneous fleet of ships to transport the cargoes, each ship with a given initial position and time for when it becomes available for new transportation tasks. Compatibility constraints between ships and cargoes may restrict which cargoes a ship can transport (e.g. due to draft limits in the ports). The shipping company sometimes also has the possibility to charter ships from the spot market to transport some of the cargoes. The planning objective in the ITSRSP is to construct routes and schedules, decide which spot cargoes to transport and which cargoes to be transported by a spot charter, so that all mandatory cargoes are transported while maximizing profit or minimizing costs. The ITSRSP extends the Pickup and Delivery Problem with Time Windows (PDPTW) with a heterogeneous fleet, compatibility constraints, different ship starting points and starting times, and service flexibility with penalties. The interplay of these complex attributes requires to jointly optimize multiple

decision sets.

These problems typically arise in the shipping of bulk products, such as crude oil, chemicals and oil products (wet bulk), and iron ore, grain, coal, bauxite, alumina and phosphate rock (dry bulk). In 2016, these product types constituted more than 60% of the weight transported in international seaborne trade. Yet, in the wake of the financial crisis in 2008, the freight rates in the dry bulk shipping segment dropped dramatically: the Baltic Dry Index dropped more than 80%, remained low since then, and experienced record lows in 2016. Additionally, for the fifth year in a row, world fleet growth has been decelerating. Despite this decline, the supply of shipping capacity still increased faster than demand, leading to a continued situation of shipping overcapacity and downward pressure on freight rates (UNCTAD, 2017). In such times, a shipping company can only obtain profit if its fleet is routed effectively.

This work is organized as follows: in Chapter 2, a formal statement is given for the ITSRSP and its extension, the Industrial and Tramp Ship Routing and Scheduling Problem with Speed Optimization (ITSRSPSO). A literature review is conducted in Chapter 3. The solution methodologies for the ITSRSP are studied in Chapter 4, and the solution methodologies for the ITSRSPSO are studied in Chapter 5. Final remarks are presented in Chapter 6.

2
Problem Statement

We now provide a formal definition for the ITSRSP and its extension, the ITSRSPSO.

2.1
The ITSRSP

The ITSRSP is defined on a complete graph $G = (V, A)$, where V is the union of a set of pickup nodes $P = \{1, \ldots, n\}$, delivery nodes $D = \{n + 1, \ldots 2n\}$, and starting locations $\{0_1, \ldots, 0_m\}$. A tramp or industrial shipping operator owns a fleet of m ships, and n cargoes are available for transportation. Each cargo $i \in \{1, \ldots, n\}$ is characterized by a load q_i and must be transported from a pickup $i \in P$ to a corresponding delivery location $n + i \in D$. Therefore, $q_i \geq 0$ for $i \in P$, and $q_i = -q_{n+i}$. Every node $i \in P \cup D$ is associated with a time window of allowable visit times $[a_i, b_i]$. Each ship $k \in \{1, \ldots, m\}$ becomes available at a time s^{D}_{0k}, at a location 0_k. It has a capacity Q_k and can traverse any arc $(i, j) \in A$ for a cost c^k_{ij} (counting fuel and canal costs) and duration δ^k_{ij}. For every ship k and node $i \in P \cup D$, there is an associated port service cost $s^{\mathrm{C}}_{ik} \geq 0$ and duration $s^{\mathrm{D}}_{ik} \geq 0$. There might be incompatibilities between ships and cargoes (e.g. due to draft limits in the ports). For each i and k, the boolean I_{ik} defines whether cargo i can be serviced by ship k. Finally, a penalty s^{C}_{i0} is paid if a cargo i is not transported by the fleet. This penalty corresponds either to the associated charter price, or the loss revenue due to not transporting an optional cargo.

The objective of ITSRSP is to form routes that minimize the sum of total travel cost and possible penalties in the case where charter ships are used or some cargoes are not transported. The routes begin at their respective starting points but have no specified endpoint, as ships operate around the clock. Every route must be feasible: ships cannot exceed their capacity, cargoes should be serviced only within their prescribed time windows, and ships cannot transport incompatible cargoes. Furthermore, routes must respect pairing and precedence constraints. The pairing constraint states that any pair $(i \in P, n+i \in D)$ must belong to the same route, and the precedence constraint states that any pickup $i \in P$ must be serviced before its delivery $n + i \in D$.

Set partitioning formulation. A simple Set Partitioning (SP) formulation of the ITSRSP is given in Eqs. (2-1) to (2-5). Let Ω_k be the set of all feasible routes for ship $k \in \{1, \ldots, m\}$. This formulation uses a binary variable $\lambda_r^k \in \Omega_k$ to indicate whether route r of vehicle k is used or not in the current solution for a cost of c_r^k. Moreover, a_{ri}^k is a binary constant valued to 1 if and only if the route r of ship k transports cargo i, and 0 otherwise. Each variable y_i is valued to 1 if and only if the cargo i is allowed to be transported by a charter instead of being included in a route.

$$\text{Minimize} \quad \sum_{k=1}^{m} \sum_{r \in \Omega_k} c_r^k \lambda_r^k + \sum_{i \in P} s_{i0}^{\mathrm{C}} y_i \tag{2-1}$$

$$\text{subject to} \quad \sum_{r \in \Omega_k} \lambda_r^k \leq 1 \qquad k = \{1, \ldots, m\} \tag{2-2}$$

$$\sum_{k=1}^{m} \sum_{r \in \Omega_k} a_{ri}^k \lambda_r^k + y_i = 1 \qquad i \in \{1, \ldots, n\} \tag{2-3}$$

$$\lambda_r^k \in \{0, 1\} \qquad k = \{1, \ldots, m\}, r \in \Omega_k \tag{2-4}$$

$$y_i \in \{0, 1\} \qquad i \in \{1, \ldots, n\}. \tag{2-5}$$

Objective (2-1) minimizes routing and charter costs. Constraint (2-2) ensures that each ship is used at most once, and Constraint (2-3) guarantees that each cargo is either transported in a route or chartered. This formulation clearly contains an exponential number of variables due to the definition of the sets Ω_k. To circumvent this issue, it is necessary to use heuristics such as the Hybrid Genetic Search (HGS), presented in Section 4.1, or exact methods such as the Branch-and-Price (B&P), presented in Section 4.2. Moreover, the SP formulation can also be used inside the HGS to find new high-quality solutions, as we explain in Section 4.1.5.

2.2 The ITSRSPSO

The ITSRSPSO is an extension of the ITSRSP where sailing speeds are a decision variable, and travel costs depend on the ship sailing speed and payload. A ship $k \in \{1, \ldots, m\}$ can traverse an arc $(i, j)^k \in A$ at any speed v (in knots) within a feasible range $[v_{\min}^k, v_{\max}^k]$, for a cost of $F_c f_{ij}^k(v, w)$. The fuel cost per tonne is represented as F_c, w is the proportion of maximum load on board the ship before arriving node j, and $f_{ij}^k(v, w)$ is the amount of fuel (in tonnes) consumed. The objective of the ITSRSPSO remains the same as the ITSRSP objective.

3
Literature Review

Early studies about ship routing and scheduling optimization date back from the 1970-80s. In a seminar study, Ronen (1983) discusses the differences between classical vehicle routing and ship routing and lists possible explanations for the scarcity of the research at the time. The author also provides a comprehensive classification scheme for various types of ship routing and scheduling problems. Since the inception of this article, research on ship routing has flourished, as highlighted by the general survey about maritime transportation by Christiansen et al. (2007), as well as dedicated recent reviews on routing and scheduling by Christiansen et al. (2013), and Christiansen & Fagerholt (2014).

Many variations of ship routing and scheduling problems have been formulated and investigated, and these problems have generally grown in richness, complexity and accuracy over the years. To name a few, Brown et al. (1987) introduced an elastic SP model to solve a full-shipload routing and scheduling problem for a fleet of crude oil tankers. Fagerholt & Christiansen (2000*b*) proposed a dynamic programming algorithm to solve a traveling salesman problem with applications to ship scheduling subproblems. The same algorithm was later exploited by Fagerholt & Christiansen (2000*a*) to solve subproblems for a multi-ship PDPTW. A maritime PDPTW with split loads was studied by Andersson et al. (2011). The authors proposed two alternative path-flow models and an exact algorithm that generates single ship schedules a priori. Vilhelmsen et al. (2014) presented a solution method based on Column Generation (CG) to solve a tramp ship routing and scheduling problem with integrated bunker optimization.

Heuristics and metaheuristics were also applied to solve several variants of ship routing problems. Some notable examples are the multi-start local search of Brønmo et al. (2007), the unified tabu search of Korsvik et al. (2009), the large neighborhood searches of Korsvik et al. (2011) and Hemmati et al. (2014), and the HGS of Borthen et al. (2017). In the latter article, hybrid genetic algorithms were used with great success to solve a multi-period supply vessel planning problem for offshore installations. Beyond this, the Unified Hybrid Genetic Search (UHGS) methodology of Vidal et al. (2012, 2014) has led to

highly accurate solutions for a considerable number of Vehicle Routing Problem (VRP) variants, including the classical Capacitated Vehicle Routing Problem (CVRP), the Vehicle Routing Problem with Time Windows (VRPTW) (Vidal et al., 2013), several prize-collecting VRPs with profits and service selections (Vidal et al., 2016; Bulhões et al., 2018), among others. Nevertheless, this methodology has neither been extended to this date to heterogeneous fixed fleet problems, nor to pickup-and-delivery problem variants, due to the necessity of designing drastically different neighborhood search operators and ensuring proper precedence and pairing between pickups and deliveries in the crossover and split operators.

A set of publicly available benchmark instances for ITSRSPs is proposed by Hemmati et al. (2014). The authors propose a mixed-integer programming model and an Adaptive Large Neighborhood Search (ALNS) metaheuristic to solve these instances. Later, Hemmati & Hvattum (2016) evaluated the effect of randomization of ALNS components on the same benchmark instances.

The effect of oil price on the optimal speed of ships was studied by Ronen (1982). The author discussed how changes in price may lead to different routing patterns (e.g. *slow steaming*). Different speed models were surveyed by Psaraftis & Kontovas (2013). A shortest path algorithm with discretized port arrival times was introduced by Fagerholt et al. (2010) to optimize the speed of fixed shipping routes. Later, an $\mathcal{O}(n^2)$ Recursive Smoothing Algorithm (RSA) was proposed by (Norstad et al., 2011) to solve the Speed Optimization Problem (SOP) inside every move evaluation of a multi-start local search heuristic for an ITSRSPSO. However, the RSA usage is limited, as it only works if the fuel consumption functions are the same for all sailing legs. Hence, it cannot be used when fuel consumption also depends on the ship payload. Wang & Meng (2012) used piecewise linear functions to approximate a fuel consumption function on a mixed-integer nonlinear programming model. Wen et al. (2016) studied a full shipload problem with load-dependent fuel consumption and proposed a three-index mixed-integer linear programming formulation and a set packing formulation for a B&P algorithm. Fukasawa et al. (2016) adapted the ITSRSP benchmark instances of Hemmati & Hvattum (2016) and proposed a SP formulation and a branch-and-cut-and-price algorithm to solve a joint routing and speed optimization problem.

4
Hybrid and Exact methods for the ITSRSP

In a recent study, Hemmati et al. (2014) have defined a broad class of ship routing and scheduling problems, and made available a benchmark suite based on real shipping segments. Due to their combination of size, variety of decision sets and constraints, the instances obtained from this suite pose considerable challenges for heuristic and exact methods. To solve these problems, we introduce a hybrid metaheuristic and an exact branch-and-price algorithm.

4.1
Hybrid Genetic Search

To solve the Industrial and Tramp Ship Routing and Scheduling Problem (ITSRSP), we propose a Hybrid Genetic Search (HGS), a non-trivial extension of the Unified Hybrid Genetic Search (UHGS) of Vidal et al. (2014). The proposed HGS includes a set-partitioning-based large neighborhood and uses problem-tailored crossover and local search operators to cover multiple ITSRSP attributes which were not included in the original framework. Being a hybrid metaheuristic, the HGS combines the exploration capabilities of genetic algorithms with efficient local search improvement procedures. Several components of the HGS ensure a balance between solution quality and population diversity, and the local search procedure also employs granular search techniques to reduce the size of the neighborhoods. Similar heuristics were widely used to successfully solve many vehicle routing problem variants (Vidal et al., 2012, 2013, 2014; Borthen et al., 2017).

The behavior of the HGS is summarized in Algorithm 4.1.1. The HGS jointly evolves a feasible and an infeasible subpopulation. At each iteration, two parents are selected. A crossover operator is applied to these parents, generating a new offspring. This offspring is improved with local search and inserted into a subpopulation, according to its feasibility. If the offspring is infeasible, a repair procedure is executed on a copy of it aiming to reach feasibility, finally inserting the resulting offspring into its subpopulation. Whenever a subpopulation reaches a maximum size, survivor selection removes individuals according to their fitness. If no improving solution is found after

a fixed number of iterations, new individuals are added to the population in order to diversify it. Penalty coefficients are periodically adjusted to control the proportion of feasible individuals generated.

Algorithm 4.1.1: HGS

```
Initialize population;
while number of iterations without improvement < It_NI and
 time < T_max do
    Select parent solutions P_1 and P_2;
    Generate offspring C from P_1 and P_2 (crossover);
    Educate offspring C (local search);
    Insert C into respective subpopulation;
    if C is infeasible then
        With probability p_rep, repair C (local search) and
        insert it into respective subpopulation;
    end
    if maximum subpopulation size reached then
        Select survivors;
    end
    if best solution not improved for It_DIV iterations then
        Diversify population;
    end
    if best solution not improved for It_SP iterations then
        Run set partitioning;
    end
    Adjust penalty coefficients for infeasibility;
end
Return best feasible solution;
```


4.1.1 Search space

The search space of the HGS includes penalized infeasible solutions. Previous work demonstrates that the use of penalized infeasible solutions enhances the search towards high-quality feasible solutions (Glover & Hao, 2009; Vidal et al., 2015*a*). A solution is infeasible if either (a) the current load of a ship exceeds its capacity at any point in the route, (b) some cargoes are not picked up or delivered within their respective time windows, or (c) a ship transports incompatible cargoes. Infeasibility (a) is penalized according to the excess of peak load in the trip. To allow time window infeasible solutions, we use the approach of Nagata et al. (2010) where time precedence constraints are relaxed. Then, the infeasibility (b) penalty is proportional to the amount of returns in time in the trip. Finally, infeasibility (c) is penalized according to the number of incompatible cargoes in the trip. No component of the HGS

allows solutions that break precedence and pairing constraints. A route r with ship k is characterized with:

$$\textit{Travel cost: } C_k(r) = \sum_{i=1}^{|r|-1} (c^k_{r_i,r_{i+1}} + s^{\mathrm{C}}_{r_i}) \tag{4-1}$$

$$\textit{Peak load: } Q^{\max}_k(r) = \max_{1\le i\le j\le |r|} \sum_{l=i}^{j} q_{r_l} \tag{4-2}$$

$$\textit{Time warp use: } TW_k(r) = \sum_{i=2}^{|r|} \max\{t^k_{r_{i-1}} + s^{\mathrm{D}}_{r_{i-1}} + \delta^k_{r_{i-1},r_i} - t^k_{r_i}, 0\} \tag{4-3}$$

$$\textit{Incompatibilities: } I_k(r) = \sum_{i=1}^{|r|} I_{r_i,k}, \tag{4-4}$$

where $t^k_{r_i}$ represents the start-of-service time at the ith node of route r with ship k, defined as

$$t^k_{r_i} = \begin{cases} a_{r_i} \ \textit{if } i = 1, \\ \max\{a_{r_i}, \min\{t_{r_{i-1}} + s^{\mathrm{D}}_{r_{i-1}} + \delta^k_{r_{i-1},r_i}, b_{r_i}\}\} \ \textit{otherwise.} \end{cases} \tag{4-5}$$

The cost of a route r with ship k equals to its travel cost $C_k(r)$. As we penalize violated constraints, we define the penalized cost this route as

$$\phi(r) = C_k(r) + \omega^{Q^{\max}} \max\{0, Q^{\max}_k(r) - Q_k\} + \omega^{TW} TW_k(r) + \omega^I I_k(r), \tag{4-6}$$

where $\omega^{Q^{\max}}$, ω^{TW} and ω^I are the respective penalty coefficients for peak load, time window, and incompatibility constraints violations. The penalized cost of a solution S is the sum of the penalized cost of all its routes, that is, $\phi(S) = \sum_{r\in\mathcal{R}^S} \phi(r)$. Penalty coefficients are adjusted during the execution of the HGS as described in Section 4.1.5.

4.1.2 Solution representation and evaluation

A solution S is represented as a giant tour π^S that holds a permutation of nodes $P \cup D$. We use a polynomial Split algorithm to obtain the routes $\mathcal{R}^S$ from π^S after crossover. Split is a dynamic programming algorithm that finds a minimum cost segmentation of a giant tour (Prins, 2004). Split was originally conceived for the Capacitated Vehicle Routing Problem (CVRP), but is flexible enough to be adapted to problem variants with pickups-and-deliveries (Velasco et al., 2009), fixed fleet size (Vidal, 2016), and heterogeneous fleet (Prins, 2009).

When dealing with vehicle routing problems with heterogeneous fleets,

several previous authors have assumed that the SPLIT algorithm should jointly optimize the giant tour segmentation and the choice of ship for each route (Duhamel et al., 2011, 2013). This extension, unfortunately, leads to a special case of a resource constrained shortest path problem, and only pseudo-polynomial algorithms are known to this date. To avoid this issue, we fix the sequence of ships and restrict the use of the SPLIT algorithm to the segmentation of the tours, in which case the ships are considered one by one in their order of appearance. To avoid any potential bias from the input instance, we shuffle the order of the ships in the solution representation. The decisions related to the use of spot charter or to not transport optional cargoes are integrated into the HGS with a dummy ship $k = 0$ that accounts for the associated penalties. This ship is the last one in the SPLIT ship permutation.

The SPLIT graph is defined as follows. Let G^S be a directed acyclic graph with nodes $V^S = \{v_0^0, \ldots, v_{2n}^0, v_0^1, \ldots v_{2n}^1, \ldots, v_0^m, \ldots, v_{2n}^m\}$ and arcs $A^S = \{(v_i^k, v_j^{k+1}) : 0 \leq i \leq j \leq 2n, 0 \leq k \leq m+1\}$ An arc $(v_i^k, v_j^{k+1}) \in A^S$ represents the route

$$r_{i+1,j}^k = (0_k, \pi_{i+1}^S, \pi_{i+2}^S, \ldots, \pi_{j-1}^S, \pi_j^S), \tag{4-7}$$

with ship k. If $i = j$, then $r_{i+1,j}^k = (0_k)$, that is, an empty route. The arc (v_i^k, v_j^{k+1}) has cost $\phi(r_{i+1,j}^k)$. If $r_{i+1,j}^k$ is not pairing or precedence feasible, then the cost of (v_i^k, v_j^{k+1}) is $+\infty$. The optimal segmentation of the giant tour is given by the shortest path between nodes v_0^0 and v_{2n}^m. The full algorithm can be implemented with time complexity $\mathcal{O}(mn^2)$ and space complexity $\mathcal{O}(mn)$.

Individual evaluation. The quality of an individual S is measured as a combination of its solution quality and diversity contribution to the subpopulation. This combination is referred to as the *biased fitness* of S in its subpopulation $\mathcal{P}$, and is defined as

$$f_{\mathcal{P}}(S) = f_{\mathcal{P}}^{\phi}(S) + \left(1 - \frac{\mu^{ELITE}}{|\mathcal{P}|}\right) f_{\mathcal{P}}^{DIV}(S), \tag{4-8}$$

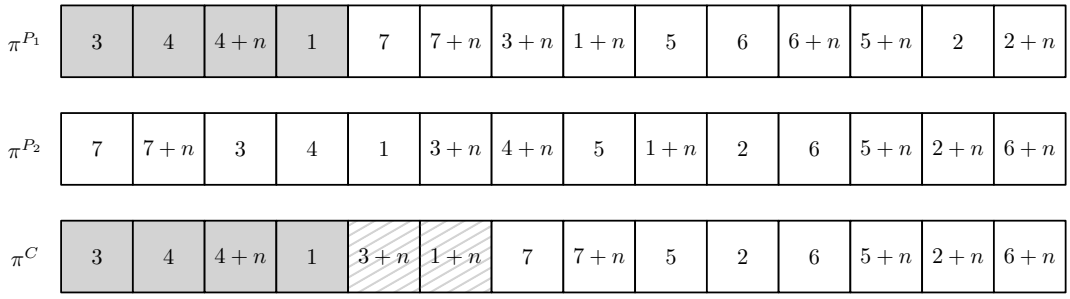
where $f_{\mathcal{P}}^{\phi}(S)$ is the penalized cost rank of S in $\mathcal{P}$, and $f_{\mathcal{P}}^{DIV}(S)$ is the diversity contribution rank of S in $\mathcal{P}$. Both ranks are relative to the subpopulation size. A parameter μ^{ELITE} balances the weight of each rank. The diversity contribution of S in $\mathcal{P}$ is defined as the average distance to its μ^{CLOSE} closest individuals. The distance metric used is the *broken pairs* distance (Campos et al., 2005; Prins, 2004): it is the proportion of arcs $(i,j)^k \in A^+$ that are in $\mathcal{R}^S$ but are not in $\mathcal{R}(S')$, where $A^+ = \{(i,j)^k \in A, k \neq 0\}$.

4.1.3 Parent selection and crossover

Two parents P_1 and P_2 are selected for crossover. Each one is obtained by a binary tournament: two individuals are selected randomly from the union of the feasible and infeasible subpopulations, and the one with smaller biased fitness is kept. We generate an offspring C using a one-point crossover operator between giant tours π^{P_1} and π^{P_2} Velasco et al. (2009).

The one-point crossover. Our one-point crossover is defined as follows: a cutting point $s \in \{1, \ldots, 2n\}$ is randomly selected, and the offspring giant tour π^C is initialized as an empty sequence. Then, nodes $\pi_1^{P_1}, \ldots, \pi_s^{P_1}$ are inserted at the end of π^C. Afterwards, the sequence of missing deliveries in $\pi_1^C, \ldots, \pi_s^C$ is inserted at the end of π^C. Finally, π^{P_2} is swept left to right, inserting missing nodes at the end of π^C. This crossover operator never generates a giant tour with a delivery before its respective pickup. After crossover, routes $\mathcal{R}^S$ are generated by Split and are optimized through education and repair. Fig. 4.1 illustrates our one-point crossover with a cutting point $s = 4$. The nodes that were copied from π^{P_1} have a gray background, missing deliveries that were inserted have a dashed background, and the remaining nodes copied from π^{P_2} have a white background.

Figure 4.1: One-point crossover illustration with a cutting point $s = 4$.



4.1.4 Education and repair

The HGS educates and repairs individuals with a first improvement local search procedure with constant time move evaluations. Routes are initially obtained with Split on a giant tour π^S, and are improved with a set of neighborhood structures evaluated in random order. When the search terminates, π^S is updated with the concatenation of all routes in $\mathcal{R}^S$. As a penalized search space is considered, solutions may be infeasible at the end of a local search. If an infeasible solution is obtained, it undergoes a *repair* phase with probability p_{rep}. Repair consists of temporarily multiplying the

base penalty coefficients by 10 and running a local search. If the resulting solution still is not feasible, then the base coefficients are multiplied by 1000 and the local search is run again.

Neighborhood structures. We use the following neighborhood structures, with size $\mathcal{O}(n^2)$:

- $\mathcal{N}_1$ (relocate pickup, intra-route only) relocate a pickup $i \in P$ after a node $j \in V$.
- $\mathcal{N}_2$ (relocate delivery, intra-route only) relocate a delivery $i \in D$ after a node $j \in V$.
- $\mathcal{N}_3^{\Delta}$ (relocate pair) relocate a pair $(i, n+i)$ placing i after a node $j \in V$ and placing $n+i$ Δ nodes after i. Restricted to $\Delta \in \{0, 1, 2\}$.
- $\mathcal{N}_4$ (swap pair) given two pairs $(i, n+i)$ and $(j, n+j)$, swap i with j and $n+i$ with $n+j$.
- $\mathcal{N}_5$ (swap ships, inter-route only) exchange the ship types between two routes.

Except for $\mathcal{N}_1$ and $\mathcal{N}_2$, the solutions found in these neighborhoods never violate pairing and precedence constraints. The remaining scenarios that break these constraints are always ignored during the local search.

Move evaluations. We perform move evaluations in $\mathcal{O}(1)$. This is possible because we represent moves as the concatenation of a constant number of route subsequences with preprocessed information. These concepts are formalized in Kindervater & Savelsbergh (1997); Irnich (2008), and Vidal et al. (2015*b*).

The subsequence resources are expressed in Table 4.1, both for subsequences with a single node (σ^0), and the concatenation of two subsequences σ and σ', according to a ship k. Given a solution S, for every subsequence $\sigma = (r_i^k, \ldots, r_j^k), 1 \leq i \leq j \leq |r^k|$ of any route $r^k \in \mathcal{R}^S$, we preprocess all resources for σ with ship k. Then, for every prefix $\sigma^p = (0^k, \ldots, r_j^k), 1 \leq j \leq |r^k|$ and suffix $\sigma^s = (r_i^k, \ldots, r_{|r^k|}^k), 1 \leq i \leq |r^k|$ subsequence of any route $r^k \in \mathcal{R}^S$, we preprocess all resources for σ^p and σ^s, for all m ships.

Memories. The HGS avoids redundant move evaluations by checking the last-modified time of a route and the last-evaluated time for each move. This constant-time check avoids the computational effort of evaluating moves that will not render an improvement.

Granular search. Our HGS follows similar granular search principles as Toth & Vigo (2003); Vidal et al. (2012), and Vidal et al. (2013), restricting move evaluations to those that create at least one directed arc (i, j) such that

Table 4.1: Subsequence resources.

Resource name	Base case	Concatenation
Travel cost	$C_k(\sigma^0) = s^{\mathrm{c}}_{ik}$	$C_k(\sigma \oplus \sigma') = C_k(\sigma) + C_k(\sigma') + c^k_{\sigma_{\lvert\sigma\rvert}\sigma'_1}$
Load	$Q_k(\sigma^0) = q_i$	$Q_k(\sigma \oplus \sigma') = Q_k(\sigma) + Q_k(\sigma')$
Peak load	$Q^{\max}_k(\sigma^0) = q_i$	$Q^{\max}_k(\sigma \oplus \sigma') = \max\{Q^{\max}_k(\sigma), Q_k(\sigma) + Q^{\max}_k(\sigma')\}$
Time warp use	$TW_k(\sigma^0) = 0$	$TW_k(\sigma \oplus \sigma') = TW_k(\sigma) + TW_k(\sigma') + \Delta^k_{TW}$
Earliest possible completion time	$E_k(\sigma^0) = a_i$	$E_k(\sigma \oplus \sigma') = \max\{E_k(\sigma') - \Delta^k, E_k(\sigma)\} - \Delta^k_{WT}$
Latest feasible starting time	$L_k(\sigma^0) = b_i$	$L_k(\sigma \oplus \sigma') = \min\{L_k(\sigma') - \Delta^k, L_k(\sigma)\} + \Delta^k_{TW}$
Duration	$D_k(\sigma^0) = s^{\mathrm{D}}_{ik}$	$D_k(\sigma \oplus \sigma') = D_k(\sigma) + D_k(\sigma') + \delta^k_{\sigma_{\lvert\sigma\rvert}\sigma'_1} + \Delta^k_{WT}$
Incompatibilities	$I_k(\sigma^0) = I_{ik}$	$I_k(\sigma \oplus \sigma') = I_k(\sigma) + I_k(\sigma')$
		$\Delta^k = D_k(\sigma) - TW_k(\sigma) + \delta^k_{\sigma_{\lvert\sigma\rvert}\sigma'_1}$
		$\Delta^k_{WT} = \max\{E_k(\sigma') - \Delta^k - L_k(\sigma), 0\}$
		$\Delta^k_{TW} = \max\{E_k(\sigma) + \Delta^k - L_k(\sigma'), 0\}$

either $i = 0$ (any ship initial location) or $j \in \Gamma(i)$. The set $\Gamma(i)$ has the $|\Gamma|$ most promising successors of i. To reasonably evaluate how good a successor is, we use a metric $\gamma(i, j)$ that considers spatial and temporal proximity between nodes. Temporal proximity is evaluated trough minimum waiting time and minimum time warp terms, with respective weight parameters γ^{WT} and γ^{TW}. We base our metric on Vidal et al. (2013) and adapt it for our application: due to the heterogeneous fleet, values are averaged over all ships, and travel cost and time are scaled since they are in different units. The metric is defined as

$$\gamma(i,j) = (\bar{c}/\bar{\delta})\bar{c}_{ij} + \gamma^{WT}\max\{a_j - \bar{s}^{\mathrm{D}}_{\boxdot i} - \bar{\delta}_{ij} - b_i, 0\} + \gamma^{TW}\max\{a_i + \bar{s}^{\mathrm{D}}_{\boxdot i} + \bar{\delta}_{ij} - b_j, 0\}, \tag{4-9}$$

where $\bar{c} = \frac{\sum_{(i,j)^k \in A^+} c^k_{ij}}{|A^+|}$, $\bar{\delta} = \frac{\sum_{(i,j)^k \in A^+} \delta^k_{ij}}{|A^+|}$, $\bar{c}_{ij} = \frac{\sum^m_{k=1} c^k_{ij}}{m}$, $\bar{s}^{\mathrm{D}}_{\boxdot i} = \frac{\sum^m_{k=1} \bar{s}^{\mathrm{D}}_{ik}}{m}$, and $\bar{\delta}_{ij} = \frac{\sum^m_{k=1} \delta^k_{ij}}{m}$.

Some edge cases are removed during the successor list construction to avoid breaking precedence constraints or to avoid the introduction of moves that create unnecessary infeasibilities:

- $n + i$ is never a successor of 0, $\forall n + i \in D$.
- i is never a successor of $n + i$, $\forall i \in P$.
- i is never a successor of j if $q_j + q_i > \max_{k=1,\ldots,m} Q_k, \forall (i, j) \in (P \cup D)^2, i \neq j$.
- $n + i$ is never a successor of j if $q_i + q_j > \max_{k=1,\ldots,m} Q_k, \forall (i, j) \in P^2, i \neq j$.
- i is never a successor of j if $\nexists k \mid I_{ik} = I_{jk} = 1, \forall (i, j) \in (P \cup D)^2, i \neq j$.

4.1.5 Population management

We implement survivor selection, population diversification, and adaptive penalties mechanisms to ensure high population diversity along with high

solution quality during the execution of the HGS. A large SP-based neighborhood is explored such that new feasible individuals are generated based on the combination of previously found feasible routes.

Initialization. To initialize the population, the HGS generates $4\mu^{MIN}$ random individuals. Random individuals are generated from giant tours where a sequence of pickups is shuffled and then deliveries are placed immediately after their respective pickups. These individuals are educated, possibly repaired and inserted into their respective subpopulation.

Survivor selection. A survivor selection mechanism occurs whenever a subpopulation reaches the maximum size of $\mu^{MIN} + \mu^{GEN}$ individuals. The μ^{GEN} individuals with maximum biased fitness are removed, considering first all individuals that have a clone. An individual P has a clone if another individual Q exists such that either (a) $\phi(P) = \phi(Q)$ or (b) $d(P,Q) = 0 \vee d(Q,P) = 0$. This survivor selection procedure has the property that, after clone removal, the best μ^{ELITE} individuals with respect to penalized cost are not discarded (Vidal et al., 2012). Thus, it establishes a balance between elitism and diversity.

Diversification. The population is diversified if no new best-known solution (BKS) was found during the last $It_{DIV} = 0.4 \cdot It_{NI}$ iterations. First, we discard all but the $\mu^{MIN}/3$ individuals with smaller biased fitness in each subpopulation. Then, we generate $4\mu^{MIN}$ new random individuals that are educated, possibly repaired and inserted in their subpopulation.

Set partitioning. To improve the solution quality of the HGS, a set-partitioning-based large neighborhood is explored to find new improving solutions based on previously found routes. This approach is inspired by the work of Subramanian et al. (2013). During the execution of the HGS, all feasible routes from generated individuals are added to the Set Partitioning (SP) formulation (Eqs. (2-1) to (2-5)). The SP is solved if no new BKS was found during the last $It_{SP} = 0.2 \cdot It_{NI}$ iterations. If the best solution found improves the BKS of the HGS, then it is inserted into the population. The SP large neighborhood provides the means to generate new solutions with an entirely different approach: instead of manipulating small segments of routes during local search, the SP neighborhood finds good combinations of previously found routes. The execution time limit of the SP is defined with a parameter $T^{SP}_{\max}$.

Penalty coefficient adjustment. Penalty coefficients are periodically adjusted to control the proportion of partially feasible individuals with respect to

each relaxed constraint. For each relaxed constraint X, we keep the proportion ξ^X of partially feasible individuals with respect to X in the last 100 iterations. This proportion does not include individuals generated by *repair* and *SP*. These coefficients are adjusted every $It_{NI}/100$ iterations: if $\xi^X \leq \xi^{REF} - 5\%$, ξ^X is multiplied by 1.2. Otherwise, if $\xi^X \geq \xi^{REF} + 5\%$, ξ^X is multiplied by 0.85. Whenever penalty coefficients are updated, each individual penalized cost and biased fitness is updated accordingly.

4.2 Branch-and-Price

Since the work of Christofides et al. (1981), Column Generation (CG) can be considered the most successful exact approach used to solve routing problems in the last years (Fukasawa et al., 2006; Ropke & Cordeau, 2009; Contardo & Martinelli, 2014). It solves the linear relaxation of a formulation with an exponential number of variables (like our SP formulation of Chapter 2) by iteratively generating the variables solving a pricing subproblem. This approach usually results in linear relaxation bounds stronger than the ones obtained by other formulations, such as flow formulations. When the CG is combined with a branch-and-bound algorithm to obtain integer solutions, it is called Branch-and-Price (B&P). In the following, we present how to consider the different attributes of the ITSRSP for both the CG and the B&P.

4.2.1 Column generation

The CG algorithm is used to avoid the enumeration of all variables of the SP formulation. Therefore, at each iteration, after solving the linear relaxation of the Master Problem (2-1)-(2-5), it calls a pricing subproblem to check the existence of a variable with negative reduced cost. In routing problems with a homogeneous fleet, the same pricing subproblem can be solved for all vehicles. This is not the case for the ITSRSP, as each ship may have a different starting location, cargo compatibility, capacity, travel costs and times. For this reason, to obtain the optimal solution of the linear relaxation, we must solve one pricing subproblem for each ship.

Let γ_k and β_i be the dual variables associated to Constraints (2-2) and (2-3), respectively. Given a^k_{ri}, a binary constant representing whether route r of ship k transports cargo i, the reduced cost of a route is defined in Equation (4-10). By decomposing this equation, we obtain the reduced cost

for each arc, as shown in Equation (4-11).

$$\bar{c}_r^k = c_r^k - \gamma_k - \sum_{k=1}^{m} a_{ri}^k \beta_i \qquad \forall k \in \{1,\ldots,m\}, r \in \Omega_k. \tag{4-10}$$

$$\bar{c}_{ij}^k = \begin{cases} c_{ij}^k - \gamma_k & \forall k \in \{1,\ldots,m\}, i = 0, j \in V, \\ c_{ij}^k - \beta_i & \forall k \in \{1,\ldots,m\}, i \in P, j \in V, \\ c_{ij}^k & \forall k \in \{1,\ldots,m\}, i \in D, j \in V. \end{cases} \tag{4-11}$$

Pricing. The pricing subproblem is an Elementary Shortest Path Problem with Resource Constraints (ESPPRC). As being an $\mathcal{NP}$-Hard problem, it is often prohibitively difficult to solve it for large instances. Several works in the literature present ways to improve its resolution mainly using route relaxation techniques (Irnich & Villeneuve, 2006; Baldacci et al., 2011). As we further discuss, we do not relax the definition of routes used in the SP formulation, but we use the Decremental State-Space Relaxation (DSSR) of Righini & Salani (2008) in the pricing subproblem.

The ESPPRC is solved using a forward dynamic programming algorithm which starts at the depot and at each iteration extends a partial path to a vertex until no further extensions are possible. For each partial path $\mathcal{P}$, we define a label $\mathcal{L}(\mathcal{P}) = (v(\mathcal{P}), \bar{c}(\mathcal{P}), q(\mathcal{P}), t(\mathcal{P}), \mathcal{O}(\mathcal{P}), \mathcal{U}(\mathcal{P}))$ containing the last vertex of the path, the accumulated reduced cost, the total load, the total time, the set of opened pairs and the set of unreachable pairs, respectively. This definition follows the one used by Ropke & Cordeau (2009) for the Pickup and Delivery Problem with Time Windows (PDPTW). Extending a path $\mathcal{P}$ to a vertex $j \in V$ is allowed only if $q(\mathcal{P}) + q_j \leq Q_k$, $t(\mathcal{P}) + \delta_{ij}^k \leq b_j$ and:

$$\begin{cases} j \notin \mathcal{U}(\mathcal{P}) & \text{if } j \in P, \\ j - n \in \mathcal{O}(\mathcal{P}) & \text{if } j \in D. \end{cases} \tag{4-12}$$

The extension being allowed, it generates the new label presented in Equation (4-13).

$$\mathcal{L}(\mathcal{P}') = \begin{cases} (j, \bar{c}(\mathcal{P}) + \bar{c}_{ij}^k, q(\mathcal{P}) + q_j, t(\mathcal{P}) + \delta_{ij}^k, \mathcal{O}(\mathcal{P}) \cup \{j\}, \mathcal{U}(\mathcal{P}) \cup \{j\}) & \text{if } j \in P, \\ (j, \bar{c}(\mathcal{P}) + \bar{c}_{ij}^k, q(\mathcal{P}) + q_j, t(\mathcal{P}) + \delta_{ij}^k, \mathcal{O}(\mathcal{P}) \backslash \{j-n\}, \mathcal{U}(\mathcal{P})) & \text{if } j \in D. \end{cases} \tag{4-13}$$

To reduce the number of labels during the dynamic programming algorithm, we use the following dominance rule: a path $\mathcal{P}_1$ dominates a path $\mathcal{P}_2$

if Eqs. (4-14) to (4-18) hold.

$$v(\mathcal{P}_1) = v(\mathcal{P}_2), \tag{4-14}$$

$$\bar{c}(\mathcal{P}_1) \leq \bar{c}(\mathcal{P}_2) \tag{4-15}$$

$$t(\mathcal{P}_1) \leq t(\mathcal{P}_2), \tag{4-16}$$

$$\mathcal{O}(\mathcal{P}_1) \subseteq \mathcal{O}(\mathcal{P}_2), \tag{4-17}$$

$$\mathcal{U}(\mathcal{P}_1) \subseteq \mathcal{U}(\mathcal{P}_2). \tag{4-18}$$

There are two important aspects of the dominance rule to discuss. First, there is no need to consider the load, since Equation (4-17) will always guarantee that $q(\mathcal{P}_1) \leq q(\mathcal{P}_2)$. Second, as discussed in Ropke & Cordeau (2009), in order to Equation (4-18) be valid, the reduced costs must satisfy the delivery triangle inequality, i.e. $\bar{c}^k_{ij} \leq \bar{c}^k_{i\ell} + \bar{c}^k_{\ell j}, \forall i \in V, j \in V, \ell \in D, k \in \{1, \ldots, m\}$. Given the definition of $\bar{c}^k_{ij}$ presented in Equation (4-11), the delivery triangle inequality holds for the reduced costs whenever it holds for the original costs c^k_{ij}. When it is not the case, a way to ensure the property is presented in Ropke & Cordeau (2009). We will discuss it further in Section 4.2.2, when we will need to apply this correction.

Decremental state-space relaxation. The DSSR is an iterative technique that relaxes a set of constraints and at each iteration restores some of them based on any infeasibility found in the current solution. The goal is to obtain the optimal solution in few iterations, restoring only a small portion of the relaxed constraints. In the case of the pricing subproblem, we relax the constraint that forbids a pair $(i, n+i)$ to be visited again by considering a set $\Gamma \subseteq P$ of pickups that are allowed to be opened again. This relaxation is done by replacing Condition (4-12) with:

$$\begin{cases} j \notin \mathcal{O}(\mathcal{P}) & \text{if } j \in \Gamma, \\ j \notin \mathcal{U}(\mathcal{P}) & \text{if } j \in P \backslash \Gamma, \\ j - n \in \mathcal{O}(\mathcal{P}) & \text{if } j \in D. \end{cases} \tag{4-19}$$

This change allows a pair to repeat, forming a cycle of pairs, but the precedence and pairing constraints for each pair occurrence are still valid. The benefit of this relaxation is to enforce the dominance rule, however, this can only be obtained changing the extension of a label from Eqs. (4-13) to (4-20).

$$\mathcal{L}(\mathcal{P}') = \begin{cases} (j, \bar{c}(\mathcal{P}) + \bar{c}^k_{ij}, q(\mathcal{P}) + q_j, t(\mathcal{P}) + \delta^k_{ij}, \mathcal{O}(\mathcal{P}) \cup \{j\}, \mathcal{U}(\mathcal{P})) & \text{if } j \in \Gamma, \\ \text{(4-13)} & \text{otherwise.} \end{cases} \tag{4-20}$$

To the best of our knowledge, this approach was never tested on the

PDPTW pricing subproblem, but it is suggested by Ropke & Cordeau (2009) almost in the same manner we propose in this work, i.e. a relaxation of $\mathcal{U}(\mathcal{P})$ sets. The authors also suggest applying the DSSR to relax $\mathcal{O}(\mathcal{P})$ sets. However, we tested this approach and it resulted in a slow convergence.

Ship ordering. As mentioned before, the ITSRSP has a heterogeneous fleet of ships. The drawback of having different ship types is to run one pricing subproblem for each ship. Since most of the time is spent on solving the subproblems, it is important to obtain a way to reduce the number of calls to them. We tested different approaches, like ship grouping and different orderings, but none of them performed well. Therefore, the CG put the ships on a list using their original ordering, and at each iteration it calls the pricing subproblem of the last ship which succeeded to obtain a new route. If the current pricing is not able to generate a new route, the procedure selects the next in the list in a circular manner. When a full round of the list is performed without generating any new route, the CG stops and returns the current solution as the linear relaxation optimal.

Heuristic pricing. As a further improvement on the resolution of the CG, the algorithm starts by solving a simple and fast heuristic pricing. During the dynamic programming, it keeps only the label with the minimum reduced cost for each vertex and each time, reducing drastically the number of labels. The ship ordering is kept the same and as soon as a full round on the list fails to obtain a new route with the heuristic pricing, the CG starts to use the exact pricing and it never uses the heuristic pricing again.

Initial solution. Considering the existence of the charters variables in the SP formulation, the CG may start with empty Ω_k sets, resulting in a solution with a high objective value, using charters for all cargoes. If there were no charters in the problem, one could use two approaches. The first is to start with a feasible solution generated by a heuristic like the one presented in Section 4.1. The second one is to introduce artificial variables in the infeasible constraints and to solve a two-phase CG, which resembles the two-phase Simplex method, minimizing the infeasibility in the first phase. The second approach is used in the Branch-and-Bound of Section 4.2.2 when new constraints are introduced in the problem.

Preprocessing. Our CG performs some preprocessing procedures based on the work of Dumas et al. (1991) to remove some arcs in the pricing subproblem. These procedures are further extended to consider the different attributes of the ITSRSP.

4.2.2 Branch-and-Bound

The CG presented in the previous section is able to produce strong lower bounds for the ITSRSP. Nevertheless, it solves only the linear relaxation of the problem, not reaching the optimal solution in most cases. In the effort to obtain the integer optimal solution, one can use several known approaches, being the Branch-and-Bound (B&B) (Land & Doig, 1960) the most successful in the known literature. The approach works by solving a relaxation of the problem and recursively splitting (branching) it into smaller subproblems, adding new constraints which will gradually enforce integrality. The full enumeration is avoided by pruning some subproblems based on their bounds. As mentioned before, when the relaxation used on each node of the B&B tree (recursion tree) is a CG algorithm, the approach is called B&P.

Branching rules. Any branching rule may be created for the B&B, as long as the resulting subproblems include all integer points from the original problem. We use three branching rules, in the order given below. Moreover, for all branching rules, the priority is given to the most fractional element.

1. Branching on charters: if any y_i variable is fractional, the B&B creates two branches, $y_i = 0$ and $y_i = 1$.

2. Branching on ships: if $\sum_{r \in \Omega_k} \lambda_r^k$ is fractional for any ship k, the B&B creates two branches, $\sum_{r \in \Omega_k} \lambda_r^k = 0$ and $\sum_{r \in \Omega_k} \lambda_r^k = 1$.

3. Branching on edges for all ships: given b_{ra}^k, a binary constant representing if route r of ship k traverses arc $a = (i,j)^k$, let us define the number of times any ship traverses $a_1 = (i,j)$ or $a_2 = (j,i)$ as $x_e = \sum_{k=1}^{m} \sum_{r \in \Omega_k} (b_{ra_1}^k + b_{ra_2}^k) \lambda_r^k$. If x_e is fractional, the B&B creates two branches, $x_e = 0$ and $x_e = 1$.

Note that we also use another branching rule based on arcs for each ship. However, in our tests, the B&B never used this last branching rule, even for the instances it was able to obtain the optimal integer solution.

The first branching rule has no impact on the pricing subproblems. On the other hand, each new constraint generated by the second and third branching rules introduces a new dual variable that must be considered by the pricing subproblems, described next.

Reduced cost. Each constraint generated by the branching on ships introduces a new dual variable τ_k, which can be considered in the reduced cost of a route like γ_k variables. These dual variables do not violate the delivery triangle

inequality since the first visited node must be a pickup node. In the case of the branching on edges for all ships, each constraint introduces a dual variable ρ_e, which will subtract the right-hand side of (4-11) in the respective case. When the third case happens, i.e. $i \in D, j \in V$, this change violates the delivery triangle inequality. To circumvent this issue, we apply the method of Ropke & Cordeau (2009) to fix the delivery triangle inequality.

Artificial variables. The branching rules presented in this section may turn the solution of a child node infeasible. However, at first, it is not possible to be sure about the feasibility of the solution because it may just be missing some columns to become feasible again. For this reason, in every node of the B&B that results in an infeasible solution, the algorithm solves the CG like a two-phase Simplex method, briefly described in Section 4.2.1. It introduces an artificial variable on each branching constraint and changes the objective function to minimize the sum of all of them, thus minimizing infeasibility. As soon as the solution becomes feasible again, the artificial variables are removed and the original objective function is restored. If the CG terminates before reaching this state, the solution is then confirmed to be infeasible.

Strong branching. Usually, we define the choice of the element to branch using a simple strategy, like selecting the most fractional variable among the branching candidates. Strong branching is a technique devised to reduce the size of the B&B tree by predicting which element will result in child nodes with better solutions. Thus, after solving the CG of each B&B node, strong branching builds a set of branching candidates for one of the branching rules explained above and simulates the branching for each element by solving both child nodes. The method then keeps the best one, i.e. the branching having the best worst child node. Furthermore, a branching with one infeasible child node is always better than one with no infeasible child node, and a branching with both child nodes infeasible is immediately chosen. Strong branching may be applied to the complete set of candidates or only to some of the most fractional candidates.

Heuristic strong branching. When the relaxation used by the B&B algorithm is the regular linear relaxation solved with the Simplex method, it is often possible to perform a complete and exact strong branching. However, this is not the case when using CG, as it is prohibitively expensive to solve it several times to obtain only two child nodes. To deal with this issue, there are some alternatives, where one is to perform strong branching only to a small subset of candidates, reducing the improvements of the method. We use another alternative that performs a heuristic strong branching, executing the

CG only with the heuristic pricing. Even with the non-optimal linear solution, it gives a good prediction for the quality of each child node and can be used to compare the result of the branching candidates. When the branching candidate is chosen, the exact pricing is executed on both child nodes.

4.3 Computational Experiments

In this section, we evaluate how efficient the HGS and the B&P are on the ITSRSP instances. Additionally, we analyze the solution quality impact of the SP component on the HGS. Finally, we evaluate how the HGS performs on ITSRSP subproblems.

Computational environment. We implemented both the HGS and the B&P in C++ using double precision floating point number representation. We compiled our code with GCC 6.3.0 using the optimization flags -O3 and -march=native. We used CPLEX 12.7 to solve the integer SP inside the HGS and the B&P master problem. We conducted our experiments on a computer with an i7-3960X CPU, 64GB of RAM, and a 64 bits Ubuntu 14.04 LTS operating system.

4.3.1 Benchmark instances for the ITSRSP

We evaluated the HGS and the B&P on a benchmark suite for the ITSRSP based on real-life scenarios, presented in Hemmati et al. (2014) and currently available at `http://home.himolde.no/~hvattum/benchmarks/`. The instances are divided in four groups of size 60, according to problem topology and cargo type: *short sea mixed load (SS_MUN)*, *short sea full load (SS_FUN)*, *deep sea mixed load (DS_MUN)*, and *deep sea full load (DS_FUN)*. For each group, there are subgroups of size 5 for each problem size. The mixed load instances have up to 130 cargoes and 40 ships, while the full load instances have up to 100 cargoes and 50 ships. To this date, 123 out of 240 instances remain open, and the lower bound of eight instances are still not known.

Adaptations for full load instances. The full load instances are a special case of the ITSRSP, similar to an asymmetric vehicle routing problem, where deliveries are performed immediately after their respective pickups. To improve the HGS execution time, we disabled some components that are not necessary to solve these instances: neighborhoods $\mathcal{N}_1$ and $\mathcal{N}_2$ are not used, and neighborhood $\mathcal{N}_3^{\Delta}$ is restricted to $\Delta = 0$. The granular search is also modified to consider a more relevant search space: $i \in D$ is the only possible successor

of $i - n$, and only pickup nodes may be successors of any delivery $i \in D$.

4.3.2 Parameters for the hybrid genetic search

We based our HGS parameters on previous values found by extensive calibration experiments on similar heuristics for related vehicle routing problems (Vidal et al., 2012, 2013). This way, we mitigated the risks of overfitting our parameters on the benchmark instances. Thus, $(\mu^{MIN}, \mu^{GEN}) = (25, 40)$, $\mu^{ELITE} = 10$, $\mu^{CLOSE} = 5$, $p_{rep} = 0.5$, $\xi^{REF} = 0.2$, $|\Gamma| = 30$, and $(\gamma^{TW}, \gamma^{WT}) = (1.0, 0.2)$. Penalty coefficients were initialized as $(\omega^{Q^{\max}}, \omega^{TW}, \omega^{I}) = (\bar{c}/\bar{q}, 100, \bar{c})$, where $\bar{q} = \frac{\sum_{i \in V} |q_i|}{|V|}$. To achieve a similar execution time with previous literature, we set $(It_{NI}, T_{\max}) = (2.5 \cdot 10^3, 15min)$, and $T^{SP}_{\max} = 30s$.

4.3.3 Performance on the ITSRSP

Hybrid genetic search. We compared our HGS results with the state-of-the-art ITSRSP heuristics from Hemmati et al. (2014) (ALNS) and Hemmati & Hvattum (2016) (ALNS-1 to ALNS-6). To measure the effectiveness of the specialized SP component, we reported results for two different versions of the HGS: one without SP (HGS-NO), and one with SP (HGS-SP). The comparison with Hemmati et al. (2014) is shown in Tables 4.2 to 4.5, while the comparison with Hemmati & Hvattum (2016) is shown in Table 4.6. Results are aggregated in groups based on instance type and size, and the reported numbers represent the average of all results for each group. As Hemmati & Hvattum (2016) used the first instance of each group (HE_1) to calibrate their heuristic, these instances are not included in the comparison presented in Table 4.6. Columns "Best" and "Avg" give, respectively, the best and the average gap (in percentage) to the previously BKS (over 10 runs). Column "T" gives the average CPU time (in minutes). Detailed results for HGS-NO and HGS-SP are shown in Tables A.1 to A.4. Full results for both Hemmati et al. (2014) and Hemmati & Hvattum (2016) were privately provided by the authors.

Both the HGS-NO and the HGS-SP largely outperform all previous heuristics, both in solution quality and execution time, becoming the new state-of-the-art heuristic for the ITSRSP. Full load instances are considerably easier to solve than mixed load instances, since the problem simplifies. The SP component of HGS-SP had a huge impact on solution quality: the average gap dropped from -0.03% to -0.36%. An interesting observation is that the

average execution time of the HGS-SP (1.22 min) is smaller than the average execution time of the HGS-NO (1.60 min). A possible explanation for this is that the SP component may contribute to bigger improvements earlier, whereas the HGS-NO may spend more time to optimize the complex decisions necessary to find these solutions.

Table 4.2: Comparison with Hemmati et al. (2014) on SS_MUN instances.

	ALNS			**HGS-NO**			**HGS-SP**		
Group	Best	Avg	T	Best	Avg	T	Best	Avg	T
SS_MUN_C7_V3	0.00	0.00	0.03	0.00	0.00	0.02	0.00	0.00	0.02
SS_MUN_C10_V3	0.00	0.00	0.04	0.00	0.00	0.03	0.00	0.00	0.03
SS_MUN_C15_V4	0.00	0.58	0.09	0.00	0.00	0.07	0.00	0.00	0.07
SS_MUN_C18_V5	0.00	0.51	0.13	0.00	0.09	0.11	0.00	0.01	0.11
SS_MUN_C22_V6	0.00	1.82	0.19	0.00	0.00	0.13	0.00	0.00	0.14
SS_MUN_C23_V13	0.00	0.57	0.25	0.00	0.06	0.18	0.00	0.00	0.17
SS_MUN_C30_V6	0.12	1.60	0.38	0.00	0.00	0.29	0.00	0.00	0.26
SS_MUN_C35_V7	0.05	1.71	0.54	-0.08	0.21	0.44	-0.20	-0.19	0.41
SS_MUN_C60_V13	0.13	1.05	2.01	-0.51	-0.15	1.68	-0.64	-0.59	1.51
SS_MUN_C80_V20	0.00	0.91	4.13	-0.98	-0.32	4.39	-1.50	-1.49	2.99
SS_MUN_C100_V30	0.00	0.95	7.77	-1.16	-0.50	7.95	-1.69	-1.68	3.93
SS_MUN_C130_V40	0.00	0.70	16.95	-0.70	-0.30	13.31	-1.82	-1.82	10.12
Avg	0.03	0.87	2.71	-0.29	-0.08	2.38	-0.49	-0.48	1.65

Table 4.3: Comparison with Hemmati et al. (2014) on SS_FUN instances.

	ALNS			**HGS-NO**			**HGS-SP**		
Group	Best	Avg	T	Best	Avg	T	Best	Avg	T
SS_FUN_C8_V3	0.00	0.00	0.03	0.00	0.00	0.01	0.00	0.00	0.01
SS_FUN_C11_V4	0.00	0.13	0.05	0.00	0.00	0.02	0.00	0.00	0.02
SS_FUN_C13_V5	0.00	0.07	0.07	0.00	0.00	0.02	0.00	0.00	0.02
SS_FUN_C16_V6	0.00	0.05	0.10	0.00	0.00	0.03	0.00	0.00	0.03
SS_FUN_C17_V13	0.01	0.01	0.14	0.00	0.00	0.04	0.00	0.00	0.04
SS_FUN_C20_V6	0.00	0.14	0.18	0.00	0.00	0.04	0.00	0.00	0.04
SS_FUN_C25_V7	0.00	0.22	0.27	0.00	0.00	0.06	0.00	0.00	0.07
SS_FUN_C35_V13	0.00	0.29	0.60	0.00	0.00	0.16	0.00	0.00	0.16
SS_FUN_C50_V20	0.00	0.35	1.38	-0.13	-0.01	0.51	-0.17	-0.17	0.38
SS_FUN_C70_V30	0.00	0.70	3.51	-0.30	-0.07	1.43	-0.58	-0.58	0.94
SS_FUN_C90_V40	0.00	0.47	6.98	-0.67	-0.42	3.37	-0.97	-0.97	1.68
SS_FUN_C100_V50	0.00	0.35	9.79	-0.40	-0.27	4.57	-0.60	-0.60	2.25
Avg	0.00	0.23	1.93	-0.12	-0.06	0.86	-0.19	-0.19	0.47

Table 4.4: Comparison with Hemmati et al. (2014) on DS_MUN instances.

	ALNS			**HGS-NO**			**HGS-SP**		
Group	Best	Avg	T	Best	Avg	T	Best	Avg	T
DS_MUN_C7_V3	0.00	0.00	0.03	0.00	0.00	0.02	0.00	0.00	0.02
DS_MUN_C10_V3	0.00	0.01	0.04	0.00	0.00	0.03	0.00	0.00	0.03
DS_MUN_C15_V4	0.00	1.26	0.08	0.00	0.00	0.07	0.00	0.00	0.06
DS_MUN_C18_V5	0.00	0.47	0.13	0.00	0.00	0.10	0.00	0.00	0.10
DS_MUN_C22_V6	0.00	2.18	0.19	0.00	0.00	0.14	0.00	0.00	0.14
DS_MUN_C23_V13	0.00	0.12	0.24	0.00	0.00	0.15	0.00	0.00	0.15
DS_MUN_C30_V6	0.24	1.04	0.37	0.11	0.15	0.27	0.00	0.01	0.25
DS_MUN_C35_V7	0.09	1.07	0.51	0.08	0.13	0.38	-0.01	-0.01	0.33
DS_MUN_C60_V13	0.24	2.66	1.92	-0.60	-0.12	2.01	-1.04	-1.03	1.36
DS_MUN_C80_V20	0.00	1.83	4.26	-0.77	-0.02	3.96	-1.23	-1.21	2.87
DS_MUN_C100_V30	0.00	1.50	8.00	-0.52	0.55	7.26	-2.10	-2.10	7.38
DS_MUN_C130_V40	0.00	1.44	17.47	-0.77	0.22	12.90	-3.55	-3.51	14.26
Avg	0.05	1.13	2.77	-0.21	0.08	2.27	-0.66	-0.65	2.25

Table 4.5: Comparison with Hemmati et al. (2014) on DS_FUN instances.

	ALNS			**HGS-NO**			**HGS-SP**		
Group	Best	Avg	T	Best	Avg	T	Best	Avg	T
DS_FUN_C8_V3	0.00	0.00	0.03	0.00	0.00	0.01	0.00	0.00	0.01
DS_FUN_C11_V4	0.00	0.00	0.05	0.00	0.00	0.02	0.00	0.00	0.02
DS_FUN_C13_V5	0.00	0.00	0.06	0.00	0.00	0.02	0.00	0.00	0.02
DS_FUN_C16_V6	0.00	0.03	0.10	0.00	0.00	0.03	0.00	0.00	0.03
DS_FUN_C17_V13	0.00	0.00	0.13	0.00	0.00	0.04	0.00	0.00	0.04
DS_FUN_C20_V6	0.00	0.01	0.16	0.00	0.00	0.04	0.00	0.00	0.04
DS_FUN_C25_V7	0.00	0.41	0.26	0.00	0.00	0.06	0.00	0.00	0.07
DS_FUN_C35_V13	0.00	1.03	0.59	0.00	0.01	0.21	0.00	0.00	0.19
DS_FUN_C50_V20	0.04	0.48	1.41	-0.11	-0.01	0.58	-0.14	-0.14	0.39
DS_FUN_C70_V30	0.00	0.28	3.55	-0.23	-0.11	1.38	-0.31	-0.31	0.90
DS_FUN_C90_V40	0.00	0.60	7.01	-0.37	-0.27	3.27	-0.50	-0.50	1.95
DS_FUN_C100_V50	0.00	0.60	9.85	-0.34	-0.20	4.77	-0.46	-0.46	2.63
Avg	0.00	0.29	1.93	-0.09	-0.05	0.87	-0.12	-0.12	0.52

Table 4.6: Comparison with Hemmati & Hvattum (2016).

	ALNS-1			**ALNS-2**			**ALNS-3**			**ALNS-4**			**ALNS-5**			**ALNS-6**			**HGS-NO**			**HGS-SP**		
Group	Best	Avg	T	Best	Avg	T	Best	Avg	T	Best	Avg	T	Best	Avg	T	Best	Avg	T	Best	Avg	T	Best	Avg	T
SS_MUN_C22_V6	0.00	0.29	0.18	0.00	0.29	0.20	0.00	0.23	0.19	0.00	0.17	0.19	0.00	0.35	0.20	0.00	0.53	0.18	0.00	0.00	0.14	0.00	0.00	0.14
SS_MUN_C23_V13	0.51	0.81	0.25	0.00	0.34	0.25	0.00	0.35	0.26	0.00	0.28	0.26	0.02	0.23	0.25	0.00	0.46	0.25	0.00	0.07	0.20	0.00	0.00	0.18
SS_MUN_C30_V6	0.17	1.32	0.35	0.00	0.66	0.40	0.10	1.14	0.41	0.00	0.90	0.42	0.00	1.04	0.39	0.00	1.94	0.38	0.00	0.00	0.28	0.00	0.00	0.26
SS_MUN_C35_V7	0.65	1.40	0.51	0.13	1.51	0.58	0.36	1.69	0.58	0.17	1.21	0.58	0.02	1.03	0.55	0.29	1.97	0.53	0.14	0.37	0.44	0.00	0.01	0.38
SS_MUN_C60_V13	1.02	2.28	2.05	0.58	1.69	2.18	1.22	2.23	2.29	0.66	1.70	2.42	0.69	2.02	2.06	0.28	1.21	1.95	-0.48	-0.08	1.64	-0.59	-0.53	1.35
DS_MUN_C22_V6	0.09	0.35	0.18	0.00	0.31	0.19	0.00	0.15	0.20	0.00	0.15	0.20	0.00	0.19	0.19	0.00	1.66	0.18	0.00	0.00	0.14	0.00	0.00	0.14
DS_MUN_C23_V13	0.00	0.00	0.26	0.00	0.00	0.26	0.00	0.02	0.28	0.00	0.01	0.28	0.00	0.00	0.25	0.00	0.00	0.24	0.00	0.00	0.15	0.00	0.00	0.15
DS_MUN_C30_V6	0.38	0.67	0.36	0.06	0.34	0.42	0.13	0.36	0.42	0.00	0.34	0.42	0.30	0.42	0.39	0.16	0.58	0.37	0.13	0.18	0.28	0.00	0.01	0.26
DS_MUN_C35_V7	0.30	0.60	0.52	0.13	0.50	0.60	0.10	0.49	0.60	0.13	0.57	0.63	0.01	0.37	0.54	0.20	0.68	0.51	-0.01	0.02	0.39	-0.01	-0.01	0.32
DS_MUN_C60_V13	2.22	3.36	2.18	0.78	2.50	2.41	0.77	3.10	2.39	0.58	3.10	2.39	0.47	2.23	1.99	0.09	2.41	1.85	-0.47	0.02	2.03	-0.97	-0.97	1.22
Avg	0.53	1.11	0.68	0.17	0.81	0.75	0.27	0.98	0.76	0.15	0.84	0.78	0.15	0.79	0.68	0.10	1.14	0.64	-0.07	0.06	0.57	-0.16	-0.15	0.44

Branch-and-price. We compared our B&P results against the results obtained from the ITSRSP mathematical formulation evaluated in Hemmati et al. (2014) with a commercial solver (MIP). The CPU time of both methods was bounded to one hour. Aggregated results are shown in Tables 4.7 to 4.10. Columns "LB Imp" and "UB Imp" represent the average relative improvement of the lower and upper bound of each instance group (in percentage), column "Opt" represents the number of optimal solutions found for each instance group, and column "T" represents the average execution time (in minutes). The results found by the B&P are largely better than the ones found by previous work: huge lower and upper bound improvements are achieved, all full cargo instances are closed, and the number of open instances went from 123 to 17. Detailed results for the B&P are shown in Tables A.5 to A.8.

Table 4.7: Comparison with Hemmati et al. (2014) on SS_MUN instances.

			MIP		**B&P**	
	LB Imp	UB Imp	Opt	T	Opt	T
SS_MUN_C7_V3	0.00	0.00	5	0.00	5	0.00
SS_MUN_C10_V3	0.00	0.00	5	0.01	5	0.00
SS_MUN_C15_V4	0.00	0.00	5	1.42	5	0.00
SS_MUN_C18_V5	2.12	0.93	4	42.50	5	0.00
SS_MUN_C22_V6	12.70	2.84	1	50.63	5	0.01
SS_MUN_C23_V13	16.60	14.88	0	59.99	5	0.14
SS_MUN_C30_V6	29.56	71.08	0	59.99	5	0.10
SS_MUN_C35_V7	34.95	73.34	0	59.99	5	0.42
SS_MUN_C60_V13	37.92	76.68	0	60.01	5	14.83
SS_MUN_C80_V20	43.64	76.35	0	60.05	5	10.16
SS_MUN_C100_V30	89.43	77.89	0	60.25	2	44.03
SS_MUN_C130_V40	99.76	77.90	0	61.52	0	60.00
Average / Total	30.56	39.33	20	43.03	52	10.81

4.3.4 Hybrid genetic search adaptations and performance on related problems

We validated the HGS on the PDPTW, the Vehicle Routing Problem with Time Windows (VRPTW) and the CVRP against Ropke & Pisinger (2006), Vidal et al. (2013), and Vidal et al. (2012), respectively.

We used some specialized components to solve each problem variant. In the VRPTW and the CVRP, (a) our neighborhoods are the same as Prins (2004), (b) Split is constrained to arcs with at most twice the maximum vehicle capacity. Whenever no shortest path is found, the value is permanently scaled by 1.1, and (c) we use the classical OX crossover (Prins, 2004). Some CVRP instances have a maximum route duration constraint. We penalized the excess of route duration in the same way as the capacity violation. To achieve

Table 4.8: Comparison with Hemmati et al. (2014) on SS_FUN instances.

			MIP		**B&P**	
	LB Imp	UB Imp	Opt	T	Opt	T
SS_FUN_C8_V3	0.00	0.00	5	0.00	5	0.00
SS_FUN_C11_V4	0.00	0.00	5	0.01	5	0.00
SS_FUN_C13_V5	0.00	0.00	5	0.00	5	0.00
SS_FUN_C16_V6	0.00	0.00	5	0.01	5	0.00
SS_FUN_C17_V13	0.00	0.00	5	0.02	5	0.00
SS_FUN_C20_V6	0.00	0.00	5	0.80	5	0.00
SS_FUN_C25_V7	0.87	0.00	3	40.47	5	0.00
SS_FUN_C35_V13	8.02	0.56	0	60.00	5	0.00
SS_FUN_C50_V20	12.05	2.01	0	60.01	5	0.01
SS_FUN_C70_V30	14.24	61.02	0	60.06	5	0.03
SS_FUN_C90_V40	17.84	73.65	0	60.29	5	0.22
SS_FUN_C100_V50	22.35	73.15	0	60.85	5	1.02
Average / Total	6.28	17.53	33	28.54	60	0.11

Table 4.9: Comparison with Hemmati et al. (2014) on DS_MUN instances.

			MIP		**B&P**	
	LB Imp	UB Imp	Opt	T	Opt	T
DS_MUN_C7_V3	0.00	0.00	5	0.01	5	000
DS_MUN_C10_V3	0.00	0.00	5	0.01	5	0.00
DS_MUN_C15_V4	0.00	0.00	5	0.35	5	0.00
DS_MUN_C18_V5	1.61	0.00	4	16.32	5	0.00
DS_MUN_C22_V6	2.77	0.09	4	26.16	5	0.00
DS_MUN_C23_V13	3.76	0.09	3	26.38	5	0.00
DS_MUN_C30_V6	31.55	49.87	0	60.00	5	0.03
DS_MUN_C35_V7	31.70	52.55	0	60.01	5	0.04
DS_MUN_C60_V13	53.58	76.90	0	60.03	5	3.84
DS_MUN_C80_V20	61.56	78.17	0	60.06	4	15.10
DS_MUN_C100_V30	95.88	76.91	0	60.27	2	42.97
DS_MUN_C130_V40	99.99	79.00	0	61.29	0	60.00
Average / Total	31.87	34.46	26	35.91	51	10.16

Table 4.10: Comparison with Hemmati et al. (2014) on DS_FUN instances.

			MIP		**B&P**	
	LB Imp	UB Imp	Opt	T	Opt	T
DS_FUN_C8_V3	0.00	0.00	5	0.01	5	0.00
DS_FUN_C11_V4	0.00	0.00	5	0.01	5	0.00
DS_FUN_C13_V5	0.00	0.00	5	0.01	5	0.00
DS_FUN_C16_V6	0.00	0.00	5	0.01	5	0.00
DS_FUN_C17_V13	0.00	0.00	5	0.02	5	0.00
DS_FUN_C20_V6	0.00	0.00	5	0.02	5	0.00
DS_FUN_C25_V7	0.00	0.00	5	0.13	5	0.00
DS_FUN_C35_V13	4.18	0.27	2	45.55	5	0.00
DS_FUN_C50_V20	7.87	0.91	1	56.25	5	0.00
DS_FUN_C70_V30	10.07	1.50	0	60.07	5	0.01
DS_FUN_C90_V40	11.74	47.49	0	60.25	5	0.10
DS_FUN_C100_V50	14.99	54.60	0	60.51	5	0.03
Average / Total	4.07	8.73	38	23.57	60	0.01

similar execution time to results on previous subproblems, we set $(It_{NI}, T_{\max})$ as $(2.5 \cdot 10^3, \infty)$ for the PDPTW, $(5 \cdot 10^3, \infty)$ for the VRPTW, and $(10^4, \infty)$ for the CVRP.

Giant tour ordering. For problems with a homogeneous fleet (PDPTW, VRPTW, and CVRP), after local search, the giant tour of a solution is obtained from the ordered concatenation of routes, according to the polar angle of their centroids.

Multistage optimization for the PDPTW and the VRPTW. Traditionally, the PDPTW and VRPTW have a hierarchical objective of minimizing fleet size first and travel cost second. We used a two-stage optimization approach to address this objective, as in Vidal et al. (2013). To minimize the fleet size, we iteratively run the HGS constraining solutions to at most m routes (initially set to a big value). As soon as a feasible solution is found, m is decremented and the search starts again, until no feasible solution is found. During the fleet minimization phase, all routes are inserted into the SP formulation, regardless of feasibility, and their cost corresponds to the sum of all infeasibilities. Then, we run the second optimization stage constraining the number of routes with the best feasible value of m found. We always reuse the population from the previous HGS call. We use the same parameters for all HGS calls.

Results for the PDPTW. We compared our results against the Adaptive Large Neighborhood Search (ALNS) of Ropke & Pisinger (2006) (RP) on the Li & Lim (2003) benchmark instances. Average results over 10 runs are shown in Table 4.11, and are grouped according to instance size. Columns "CNV" and "CTD" give the cumulative number of vehicles and the cumulative travel distance, while column "T" gives the average CPU time (in minutes). On average, for instances with size 100 and 200, the HGS-SP outperformed the ALNS. For instances with size 400, the HGS-SP performed almost as well as the ALNS.

Table 4.11: Results on Li & Lim (2003) PDPTW instances.

	RP			**HGS-SP**		
n	CNV	CTD	T	CNV	CTD	T
100	403.00	58249.42	1.10	402.00	58060.15	0.49
200	608.10	181707.35	4.40	606.00	182177.89	3.08
400	1167.80	425816.87	14.68	1168.00	428989.38	19.29
	Pentium IV 1.5GHz			i7-3960X 3.3GHz		

Results for the VRPTW. We evaluated our method with the Solomon

& Desrosiers (1988) and Gehring & Homberger (1999) VRPTW benchmark instances. We compared our results against the HGSADC of Vidal et al. (2013). Average results over five runs are shown in Table 4.12, and are grouped according to instance size. Columns "CNV" and "CTD" give the cumulative number of vehicles and the cumulative travel distance, while column "T" gives the CPU time (in minutes). On average, for instances with size 100, the HGS-SP outperformed HGSADC. For instances with size 200 and 400, the HGS-SP performed almost as well as the HGSADC.

Table 4.12: Results on Solomon & Desrosiers (1988) and Gehring & Homberger (1999) VRPTW instances.

	HGSADC			**HGS-SP**		
n	CNV	CTD	T	CNV	CTD	T
100	405.00	57218.00	2.68	405.00	57209.70	1.81
200	694.00	168407.00	8.40	694.00	168585.64	7.40
400	1382.00	388697.00	34.10	1382.20	391079.52	41.30
	Xe-2.93G			i7-3960X 3.3GHz		

Results for the CVRP. We evaluated our method on Christofides et al. (1979) (*CMT*) and Golden et al. (1998) (*Golden*) CVRP benchmark instances and compared our results against the HGSADC of Vidal et al. (2012). Results over 10 runs are shown in Table 4.13. Column "Gap" gives the gap of the average solution (in percentage) to the BKS, and column "T" gives the average execution time (in minutes). The HGS-SP results are nearly as good as the HGSADC results.

Table 4.13: Results on Christofides et al. (1979) and Golden et al. (1998) CVRP instances.

	HGSADC		**HGS-SP**	
Group	Gap	T	Gap	T
CMT	0.05	2.21	0.06	3.43
Golden	0.28	28.53	0.38	22.83
Avg	0.16	15.37	0.22	13.13
	Pentium IV 3.0GHz		i7-3960X 3.3GHz	

5
Joint Ship Speed Optimization and Heuristics

Fuel consumption is a significant component of the operating costs of ships (Psaraftis & Kontovas, 2013). The fuel consumption per time unit of ships can be approximated by a cubic function of speed (Ronen, 1982), meaning that different speeds can considerably change the costs associated with fuel consumption. Speed optimization is notably relevant for the shipping industry: operators often sail at slow speeds (*slow steaming*) to minimize fuel costs when there is an oversupply of shipping capacity in the market (Ronen, 1982). However, most routing models assume a constant speed, making it impossible to efficiently explore the trade-off between routing decisions and fuel consumption. Other factors may also contribute to fuel consumption, such as the ship payload, weather conditions, and the ship machinery type (Psaraftis & Kontovas, 2013). Overall, the integration of speed decisions into the routing process and the evaluation of fuel costs based on speed (and other factors) may lead to a better fleet utilization and increased profits.

In this work, we study heuristic methods for a ship routing problem where the speed on each sailing leg (route segment) is a decision variable, and fuel consumption is a convex function of speed and payload. The studied problem is referred to as the Industrial and Tramp Ship Routing and Scheduling Problem with Speed Optimization (ITSRSPSO), and it extends the Industrial and Tramp Ship Routing and Scheduling Problem (ITSRSP) with speed decisions. The influence of payload on fuel consumption extends previous work (Norstad et al., 2011), and results in leg-dependent fuel consumption functions, as the load on board of the ship changes at every port visit.

The problem of finding the optimal speed for each segment of a route is referred to as the Speed Optimization Problem (SOP). The SOP with fuel consumption functions and time-window constraints can be viewed as a Resource Allocation Problem with Nested Constraints (RAP-NC), where a time budget is allocated on sailing legs. If the fuel consumption functions are convex, the RAP-NC can be solved in $\mathcal{O}(n \log m \log \frac{nB}{\epsilon})$ with the Monotonic Decomposition Algorithm (MDA) of Vidal et al. (2017). We propose an extension of the Hybrid Genetic Search (HGS) described in Section 4.1 for the ITSRSPSO. This extension uses the MDA to solve the SOP on every local

search move evaluation.

5.1 Hybrid Genetic Search with Optimal Speed Decisions

We propose a local search framework inside the HGS where all time-window feasible routes have optimal speed decisions. As our moves always involve one or two routes, we must solve at most two SOPs every time a move is evaluated. We discuss how to find optimal speed decisions on fixed routes, then propose lower bounds for move evaluations, and finally describe how to extend the HGS.

5.1.1 Optimization of speed decisions on a fixed route

We now consider the SOP on a fixed route $r = (r_1, \ldots, r_n)$, serviced by ship k. Let w_i be a constant that represents the proportion of maximum load on board the ship before arriving node r_i (with $w_1 = 0$). The RAP-NC formulation of the SOP is shown in Eqs. (5-1) to (5-4), where: (a) F_c is the fuel cost per tonne, (b) v_i is the speed (in knots) used to sail from node r_{i-1} to r_i, (c) ℓ^k_{ij} is the travel distance between the two nodes (in nautical miles), and (d) B is the total time budget that can be allocated, and is equal to the latest feasible arrival time at the last node of r. Next, we describe two algorithms for solving the SOP.

$$\min\ f(\mathbf{x}) = F_c \sum_{i=1}^{n} f^k_{r_{i-1},r_i}(v_i, w_i) \tag{5-1}$$

$$\text{s.t.}\quad a_i \leq \sum_{j=1}^{i} \frac{\ell^k_{r_{j-1},r_j}}{v_j} \leq b_i \qquad i \in \{1, \ldots, m-1\} \tag{5-2}$$

$$\sum_{i=1}^{n} \frac{\ell^k_{r_{i-1},r_i}}{v_i} = B \tag{5-3}$$

$$v^k_{\min} \leq v_i \leq v^k_{\max} \qquad i \in \{1, \ldots, n\}. \tag{5-4}$$

A discretized arrival times algorithm. A Discretized Arrival Times (DAT) algorithm can be used to solve the SOP on a route r (Fagerholt et al., 2010). We build a directed acyclic graph $G_r = (V_r, A_r)$ where route visits are replicated with discretized arrival times over their respective time windows. A node $(r_i, t) \in V_r$ represents an arrival at r_i at time $t \in [a_{r_i}, b_{r_i}]$. An edge $((r_{i-1}, t), (r_i, t')) \in A_r$ represents a trip from node r_{i-1} to r_i in $t' - t$ hours, at speed $v_i = \frac{\ell^k_{r_{i-1},r_i}}{t'-t}$, with a cost $F_c f^k_{r_{i-1},r_i}(v_i, w_i)$. We connect a source node to all nodes $(r_1, t), \forall t$, and all nodes $(r_{|r|}, t), \forall$ to a sink node, and obtain the shortest

path from source to sink to solve the SOP. The complexity of this algorithm is $\mathcal{O}(|V_r|^2)$, where the size of V_r depends on the number of replicated nodes.

A monotonic decomposition algorithm. The MDA is a divide-and-conquer algorithm that solves four Resource Allocation Problem (RAP) subproblems on each node of the recursion tree. The RAP formulations are equivalent to a RAP-NC due to variable bounds obtained from optimal solutions deeper into the recursion tree. To solve a RAP subproblem, we adopt a simple strategy that performs a binary search over its single dual variable (Patriksson, 2008). The MDA has a complexity of $\mathcal{O}(n \log m \log \frac{nB}{\epsilon})$, where n is the route size, m is the number of nested time-window constraints (in our case, $n = m$), B is the total time budget, and ϵ is a tolerance parameter.

Service cost and time. For a fixed route, service costs s^{C}_{ik} are always constant. Thus, they are not taken into consideration during the speed optimization. Service times can be represented by manipulating the RAP-NC formulation: for every node $r_i, i = 1, \ldots, n$, we calculate the sum of previous service times $S_{i-1} = \sum_{j=1}^{i-1} s^{\mathrm{D}}_{r_j k}$ and set $a_i \leftarrow a_i - S_{i-1}$ and $b_i \leftarrow b_i - S_{i-1}$.

Waiting times. Ships have a minimum and maximum sailing speed. Therefore, they may wait on an early arrival due to a combination of speed bounds and time windows. This leads to fuel consumption functions that are not strictly-convex ($if\ v < v^k_{\min}, f^k_{ij}(v, w) = f^k_{ij}(v^k_{\min}, w)$) in the RAP-NC formulation. Fortunately, both algorithms do not require strict convexity, and are therefore able to account for waiting times.

5.1.2 Local search

The speed of each leg is jointly optimized *on every local search move evaluation*. If a neighbor solution is time-window infeasible, the move is evaluated in $\mathcal{O}(1)$ using preprocessed information and route concatenation techniques. For this scenario, travel cost is obtained from the fuel consumption functions assuming maximum ship speed and full load. Otherwise, if a neighbor solution is time-window feasible, the optimal speed for each leg is found with the MDA.

Lower bounds on move evaluations. A significant amount of computational effort is necessary to solve the SOP on every move evaluation. To reduce this effort, we introduce a two-phase neighbor evaluation scheme. First, the move evaluation compares the cost of the current solution with a lower bound on the neighbor cost. If the lower bound is worse than the current solu-

tion, then the SOP is not solved. Otherwise, we run the MDA. We use the following lower bound: $v_i = v^k_{\min}, i = 1, \ldots, n$.

5.1.3 A hybrid genetic search extension

To solve the ITSRSPSO, we introduce an extension of the HGS described in Section 4.1. There are two components of the HGS that must be adapted: the local search used during education and repair, and the Split algorithm. We adapt the local search component with the methodology covered in Section 5.1. In the ITSRSP, the Split routes can be evaluated incrementally, thus, allowing a constant-time evaluation of route information. This is not the case for the ITSRSPSO, since the optimal speed decisions must be re-evaluated as the routes are incremented. Hence, every route evaluated by Split solves a SOP with the MDA.

5.2 Computational Experiments

We introduced a new set of ITSRSPSO instances based on the Hemmati et al. (2014) benchmark suite for the ITSRSP. Then, we compared the performance of the MDA and the DAT on fixed routes. Finally, we evaluated the HGS with joint speed optimization against a strategy that only optimizes the speed once at the end.

Computational environment. We used the same computational environment and parameters described in Section 4.3, and set a tolerance of $\epsilon = 1\text{e}{-5}$ for the MDA.

5.2.1 Benchmark instances

We generated ITSRSPSO instances based on the ITSRSP instances of Hemmati et al. (2014). As the ITSRSPSO is a heterogeneous ship routing problem, different ships may have different speed bounds and fuel consumption profiles. The type of any ship $k \in \{1, \ldots, m\}$ is specified by the instance files. This type defines the ship design speed v^k_{D} and a ship-specific coefficient μ_k, related to fuel consumption. These values are detailed in Table 5.1. Next, we describe how we adapted data from the original instances, and detail information related to speed bounds and fuel consumption.

Arc lengths. We assumed that the original ITSRSP travel time values δ^k_{ij} correspond to the time needed to sail an arc at design speed. Hence, we calculated the length of an arc $(i, j)^k \in A$ as follows: $\ell^k_{ij} = v^k_{\text{D}} \delta^k_{ij}$.

Speed bounds. We defined the minimum and maximum ship sailing speed as a proportion of its design speed. Thus, $v^k_{\min} = 0.6v^k_{\text{D}}$ and $v^k_{\max} = 1.2v^k_{\text{D}}$.

Fuel consumption functions and fuel cost. For any arc $(i,j)^k \in A$, we used fuel consumption functions in the form $f^k_{ij}(v,w) = \frac{1}{24}\mu_k v^2 \ell^k_{ij}(0.8 + 0.2w)$. We assumed a fuel cost per tonne of 590.

Table 5.1: Information for each ship type.

	Short sea		**Deep sea**	
Ship type	Design speed	Coefficient	Design speed	Coefficient
1	13.5	0.0106	14.5	0.0148
2	14.5	0.0079	13.5	0.0126
3	12.0	0.0139	24.6	0.0134
4	13.2	0.0137	14.5	0.0164
5	13.5	0.0057	25.0	0.0160
6	13.0	0.0046	14.5	0.0197
7	12.5	0.0072	15.5	0.0250
8	-	-	15.5	0.0234

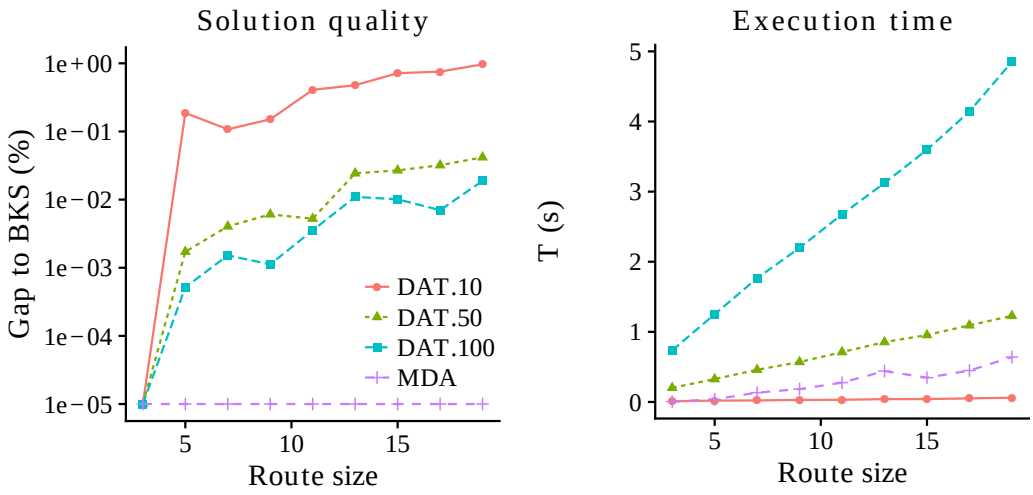
5.2.2 Evaluation of fixed routes

We evaluated both the MDA and the DAT algorithms on a set of 90 routes obtained from the ITSRSPSO instances. Route sizes vary between three and 19 nodes. There are 10 routes per instance size. We evaluated the DAT algorithm with three different discretization levels: 10, 50, and 100 (respectively, DAT-10, DAT-50, and DAT-100). Results over 5.000 runs are illustrated in Fig. 5.1 and detailed in Table 5.2. The reported numbers represent the average of all results for each instance group. Column "Gap" gives the gap to the best known solution (BKS), while column "T" gives the CPU time (in seconds). Our results show that the MDA has the best trade-off between solution quality and execution time. The solution quality of the MDA is always the best one, and its execution time is only dominated by the DAT-10. However, the DAT-10 has the worst gap to the BKS. Furthermore, as the DAT discretization level increases, the CPU time grows excessively. Therefore, the DAT is unfit for usage during local search.

5.2.3 Routing with joint speed optimization

We compared our HGS with joint speed optimization (HGS-J) against a version of the HGS that assumes a fixed speed and optimizes speed once at the

Figure 5.1: Comparison of MDA against DAT on fixed routes.



end (HGS-O). Both versions use the MDA as a SOP solver. Results are shown in Tables 5.3 to 5.6, and are grouped according to problem topology (short sea, deep sea) and cargo type (mixed load, full load). The reported numbers represent the average of all results for each instance group. Average values for the gap to the previously BKS (in percentage) and CPU time (in minutes) are respectively given by "Gap" and "T".

The average solution quality of the HGS-J is largely better than the one of HGS-O, with only a moderate increase (7.1×) of computational effort. Given the complexity of a tight integration of speed optimization and routing decisions in a metaheuristic, the observed CPU time is very satisfying. However, for instances with 30 cargoes or more, the HGS-J easily reaches the CPU time limit. Nevertheless, for most instance groups, the HGS-J results still outperform the HGS-O. But this is not the case for DS_MUN_C130_V40: the solutions found by the HGS-J have a gap of 8.78% to the BKS (found by the HGS-O). This happens because most of the CPU time is spent during the HGS-J population initialization, leaving little CPU time left to improve the solutions found. This scenario can be circumvented either by increasing the CPU time limit of the HGS-J, or by introducing further improvements into the SOP solver.

Table 5.2: Comparison of the MDA and DAT on fixed routes.

	DAT-10		**DAT-50**		**DAT-100**		**MDA**	
n	Gap	T	Gap	T	Gap	T	Gap	T
3	0.00	0.01	0.00	0.20	0.00	0.74	0.00	0.01
5	0.19	0.02	0.00	0.33	0.00	1.25	0.00	0.04
7	0.11	0.02	0.00	0.46	0.00	1.76	0.00	0.13
9	0.15	0.03	0.01	0.57	0.00	2.20	0.00	0.18
11	0.41	0.03	0.01	0.71	0.00	2.68	0.00	0.28
13	0.48	0.04	0.02	0.85	0.01	3.12	0.00	0.44
15	0.72	0.04	0.03	0.95	0.01	3.60	0.00	0.35
17	0.76	0.05	0.03	1.09	0.01	4.14	0.00	0.45
19	0.97	0.06	0.04	1.23	0.02	4.85	0.00	0.64
Avg	0.42	0.03	0.02	0.71	0.01	2.70	0.00	0.28

Table 5.3: Evaluation of the impact of joint speed optimization on ITSRSPSO SS_MUN instances.

	HGS-O			**HGS-J**		
Group	Best	Avg	T	Best	Avg	T
SS_MUN_C7_V3	1.75	1.90	0.02	0.00	0.00	0.31
SS_MUN_C10_V3	3.21	3.25	0.04	0.00	0.00	0.94
SS_MUN_C15_V4	7.63	8.21	0.08	0.00	0.00	3.64
SS_MUN_C18_V5	8.94	9.14	0.12	0.00	0.00	5.49
SS_MUN_C22_V6	8.14	8.25	0.17	0.00	0.00	8.23
SS_MUN_C23_V13	5.09	5.30	0.19	0.00	0.00	5.79
SS_MUN_C30_V6	10.07	10.24	0.30	0.00	0.15	15.00
SS_MUN_C35_V7	7.65	7.95	0.46	0.00	0.32	15.00
SS_MUN_C60_V13	9.52	10.23	1.53	0.00	1.00	15.01
SS_MUN_C80_V20	5.95	6.45	3.61	0.00	0.88	15.02
SS_MUN_C100_V30	5.38	5.75	5.76	0.00	0.28	15.02
SS_MUN_C130_V40	4.61	5.29	12.48	0.00	0.49	15.03
Avg	6.49	6.83	2.06	0.00	0.26	9.54

Table 5.4: Evaluation of the impact of joint speed optimization on ITSRSPSO SS_FUN instances.

	HGS-O			**HGS-J**		
Group	Best	Avg	T	Best	Avg	T
SS_FUN_C8_V3	3.83	3.83	0.01	0.00	0.00	0.45
SS_FUN_C11_V4	9.31	9.31	0.02	0.00	0.00	1.20
SS_FUN_C13_V5	6.28	6.65	0.03	0.00	0.00	1.41
SS_FUN_C16_V6	4.56	4.83	0.04	0.00	0.00	2.14
SS_FUN_C17_V13	6.06	6.84	0.06	0.00	0.00	1.65
SS_FUN_C20_V6	10.03	11.10	0.05	0.00	0.00	4.55
SS_FUN_C25_V7	8.61	9.08	0.08	0.00	0.00	8.17
SS_FUN_C35_V13	8.55	9.16	0.21	0.00	0.00	14.59
SS_FUN_C50_V20	10.19	10.99	0.55	0.00	0.00	15.04
SS_FUN_C70_V30	12.40	12.82	1.43	0.00	0.72	15.01
SS_FUN_C90_V40	12.26	12.92	2.78	0.00	0.28	15.01
SS_FUN_C100_V50	16.68	17.30	3.73	0.00	0.12	15.01
Avg	9.06	9.57	0.75	0.00	0.09	7.85

Table 5.5: Evaluation of the impact of joint speed optimization on ITSRSPSO DS_MUN instances.

	HGS-O			**HGS-J**		
Group	Best	Avg	T	Best	Avg	T
DS_MUN_C7_V3	4.02	4.11	0.02	0.00	0.00	0.39
DS_MUN_C10_V3	5.26	5.34	0.03	0.00	0.00	0.90
DS_MUN_C15_V4	7.79	7.84	0.07	0.00	0.00	3.86
DS_MUN_C18_V5	10.06	10.08	0.11	0.00	0.00	5.90
DS_MUN_C22_V6	17.32	17.52	0.16	0.00	0.00	9.08
DS_MUN_C23_V13	11.33	11.80	0.19	0.00	0.00	5.37
DS_MUN_C30_V6	11.86	12.81	0.27	0.00	0.07	15.06
DS_MUN_C35_V7	8.92	9.39	0.37	0.00	0.69	15.01
DS_MUN_C60_V13	8.58	9.15	1.62	0.00	1.99	15.03
DS_MUN_C80_V20	16.45	18.65	3.54	0.00	1.47	15.02
DS_MUN_C100_V30	12.45	13.38	8.36	0.00	1.04	15.03
DS_MUN_C130_V40	0.00	0.97	15.08	8.78	12.60	15.06
Avg	9.50	10.09	2.48	0.73	1.49	9.64

Table 5.6: Evaluation of the impact of joint speed optimization on ITSRSPSO DS_FUN instances.

	HGS-O			**HGS-J**		
Group	Best	Avg	T	Best	Avg	T
DS_FUN_C8_V3	8.72	8.72	0.01	0.00	0.00	0.56
DS_FUN_C11_V4	15.41	15.41	0.02	0.00	0.00	1.16
DS_FUN_C13_V5	2.82	2.92	0.03	0.00	0.00	1.62
DS_FUN_C16_V6	12.59	12.76	0.04	0.00	0.00	2.58
DS_FUN_C17_V13	10.79	11.16	0.06	0.00	0.00	2.07
DS_FUN_C20_V6	9.96	10.40	0.05	0.00	0.00	5.03
DS_FUN_C25_V7	8.73	9.60	0.08	0.00	0.00	9.86
DS_FUN_C35_V13	9.70	9.81	0.26	0.00	0.00	15.05
DS_FUN_C50_V20	18.12	18.48	0.63	0.00	0.03	15.04
DS_FUN_C70_V30	20.92	21.20	1.48	0.00	0.48	15.01
DS_FUN_C90_V40	17.15	17.56	3.23	0.00	0.64	15.02
DS_FUN_C100_V50	19.82	20.05	4.43	0.00	0.51	15.03
Avg	12.89	13.17	0.86	0.00	0.14	8.17

6
Concluding Remarks

In this work, we explored heuristic and exact solution methodologies for a routing problem with important applications in the shipping industry. We showed how successful methodologies for related routing problems can be adapted to efficiently solve ship routing problems. We conducted extensive computational experiments on a set of benchmark instances based on real shipping segments. The results found demonstrate that our methodologies largely outperform the state-of-the-art heuristic and exact methods for the ITSRSP.

Then, we extended our heuristic method to solve a ship routing problem extension where speed is a decision variable. Our results demonstrated that a joint speed and routing optimization methodology is highly profitable from an operational cost perspective. As the model becomes richer with details, it better represents reality, with the drawback of increasing the complexity to solve it. Thus, the need for efficient algorithms becomes clear, and solving such model is only made possible with the use of an efficient linearithmic algorithm such as the MDA.

A promising research venue remains open: being a divide-and-conquer algorithm, the MDA can be adapted to work on preprocessed subsequence information, similar to the concatenation-based move evaluations presented in Section 4.1.4. This may allow the pruning of the MDA recursion tree, and contribute to significant CPU time improvements.

Bibliography

Andersson, H., Christiansen, M. & Fagerholt, K. (2011), 'The maritime pickup and delivery problem with time windows and split loads', *INFOR: Information Systems and Operational Research* **49**(2), 79–91.

Baldacci, R., Mingozzi, A. & Roberti, R. (2011), 'New route relaxation and pricing strategies for the vehicle routing problem', *Operations Research* **59**(5), 1269–1283.

Borthen, T., Loennechen, H., Wang, X., Fagerholt, K. & Vidal, T. (2017), 'A genetic search-based heuristic for a fleet size and periodic routing problem with application to offshore supply planning', *EURO Journal on Transportation and Logistics*.

Brønmo, G., Christiansen, M., Fagerholt, K. & Nygreen, B. (2007), 'A multi-start local search heuristic for ship scheduling—a computational study', *Computers & Operations Research* **34**(3), 900–917.

Brown, G. G., Graves, G. W. & Ronen, D. (1987), 'Scheduling ocean transportation of crude oil', *Management Science* **33**(3), 335–346.

Bulhões, T., Hà, M., Martinelli, R. & Vidal, T. (2018), 'The vehicle routing problem with service level constraints', *European Journal of Operational Research* **265**(2), 544–558.

Campos, V., Laguna, M. & Martí, R. (2005), 'Context-independent scatter and tabu search for permutation problems', *INFORMS Journal on Computing* **17**(1), 111–122.

Christiansen, M. & Fagerholt, K. (2014), Chapter 13: Ship routing and scheduling in industrial and tramp shipping, *in* P. Toth & D. Vigo, eds, 'Vehicle Routing', Society for Industrial and Applied Mathematics, pp. 381–408.

Christiansen, M., Fagerholt, K., Nygreen, B. & Ronen, D. (2007), Chapter 4 maritime transportation, *in* C. Barnhart & G. Laporte, eds, 'Transportation', Elsevier, pp. 189–284.

Christiansen, M., Fagerholt, K., Nygreen, B. & Ronen, D. (2013), 'Ship routing and scheduling in the new millennium', *European Journal of Operational Research* **228**(3), 467–483.

Christofides, N., Mingozzi, A. & Toth, P. (1979), The vehicle routing problem, *in* N. Christofides, A. Mingozzi, P. Toth & C. Sandi, eds, 'Combinatorial Optimization', Wiley, Chichester, pp. 315–338.

Christofides, N., Mingozzi, A. & Toth, P. (1981), 'Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations', *Mathematical Programming* **20**, 255–282.

Contardo, C. & Martinelli, R. (2014), 'A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints', *Discrete Optimization* **12**, 129–146.

Duhamel, C., Gouinaud, C., Lacomme, P. & Prodhon, C. (2013), A multi-thread GRASPxELS for the heterogeneous capacitated vehicle routing problem, *in* E.-G. Talbi, ed., 'Hybrid Metaheuristics', Studies in Computational Intelligence, Springer Berlin Heidelberg, pp. 237–269.

Duhamel, C., Lacomme, P. & Prodhon, C. (2011), 'Efficient frameworks for greedy split and new depth first search split procedures for routing problems', *Computers & Operations Research* **38**(4), 723–739.

Dumas, Y., Desrosiers, J. & Soumis, F. (1991), 'The pickup and delivery problem with time windows', *European Journal of Operational Research* **54**(1), 7 – 22.

Fagerholt, K. & Christiansen, M. (2000*a*), 'A combined ship scheduling and allocation problem', *The Journal of the Operational Research Society* **51**(7), 834–842.

Fagerholt, K. & Christiansen, M. (2000*b*), 'A travelling salesman problem with allocation, time window and precedence constraints — an application to ship scheduling', *International Transactions in Operational Research* **7**(3), 231–244.

Fagerholt, K., Laporte, G. & Norstad, I. (2010), 'Reducing fuel emissions by optimizing speed on shipping routes', *Journal of the Operational Research Society* **61**(3), 523–529.

Fukasawa, R., He, Q., Santos, F. & Song, Y. (2016), 'A joint routing and speed optimization problem', *ArXiv e-prints*.

Fukasawa, R., Longo, H., Lysgaard, J., Poggi de Aragão, M., Reis, M., Uchoa, E. & Werneck, R. (2006), 'Robust branch-and-cut-and-price for the capacitated vehicle routing problem', *Mathematical Programming* **106**(3), 491–511.

Gehring, H. & Homberger, J. (1999), A parallel hybrid evolutionary metaheuristic for the vehicle routing problem with time windows, *in* 'Proceedings of EUROGEN99', Vol. 2, Springer Berlin, pp. 57–64.

Glover, F. & Hao, J.-K. (2009), 'The case for strategic oscillation', *Annals of Operations Research* **183**(1), 163–173.

Golden, B. L., Wasil, E. A., Kelly, J. P. & Chao, I.-M. (1998), The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results, *in* T. G. Crainic & G. Laporte, eds, 'Fleet Management and Logistics', Springer US, pp. 33–56.

Hemmati, A. & Hvattum, L. M. (2016), 'Evaluating the importance of randomization in adaptive large neighborhood search', *International Transactions in Operational Research* **24**(5), 929–942.

Hemmati, A., Hvattum, L. M., Fagerholt, K. & Norstad, I. (2014), 'Benchmark suite for industrial and tramp ship routing and scheduling problems', *INFOR: Information Systems and Operational Research* **52**(1), 28–38.

Irnich, S. (2008), 'A unified modeling and solution framework for vehicle routing and local search-based metaheuristics', *INFORMS Journal on Computing* **20**(2), 270–287.

Irnich, S. & Villeneuve, D. (2006), 'The shortest-path problem with resource constraints and k-cycle elimination for k $\geq$ 3', *INFORMS Journal on Computing* **18**(3), 391–406. ISSN 1526-5528.

Kindervater, G. A. & Savelsbergh, M. W. (1997), 'Vehicle routing: handling edge exchanges', *Local search in combinatorial optimization* pp. 337–360.

Korsvik, J. E., Fagerholt, K. & Laporte, G. (2009), 'A tabu search heuristic for ship routing and scheduling', *Journal of the Operational Research Society* **61**(4), 594–603.

Korsvik, J. E., Fagerholt, K. & Laporte, G. (2011), 'A large neighbourhood search heuristic for ship routing and scheduling with split loads', *Computers & Operations Research* **38**(2), 474–483.

Land, A. H. & Doig, A. G. (1960), 'An automatic method of solving discrete programming problems', *Econometrica* **28**(3), 497–520.

Li, H. & Lim, A. (2003), 'A metaheuristic for the pickup and delivery problem with time windows', *International Journal on Artificial Intelligence Tools* **12**(02), 173–186.

Nagata, Y., Bräysy, O. & Dullaert, W. (2010), 'A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows', *Computers & Operations Research* **37**(4), 724–737.

Norstad, I., Fagerholt, K. & Laporte, G. (2011), 'Tramp ship routing and scheduling with speed optimization', *Transportation Research Part C: Emerging Technologies* **19**(5), 853–865.

Patriksson, M. (2008), 'A survey on the continuous nonlinear resource allocation problem', *European Journal of Operational Research* **185**(1), 1–46.

Prins, C. (2004), 'A simple and effective evolutionary algorithm for the vehicle routing problem', *Computers & Operations Research* **31**(12), 1985 – 2002.

Prins, C. (2009), 'Two memetic algorithms for heterogeneous fleet vehicle routing problems', *Engineering Applications of Artificial Intelligence* **22**(6), 916 – 928. Artificial Intelligence Techniques for Supply Chain Management.

Psaraftis, H. N. & Kontovas, C. A. (2013), 'Speed models for energy-efficient maritime transportation: a taxonomy and survey', *Transportation Research Part C: Emerging Technologies* **26**, 331–351.

Righini, G. & Salani, M. (2008), 'New dynamic programming algorithms for the resource constrained elementary shortest path problem', *Networks* **51**(3), 155–170.

Ronen, D. (1982), 'The effect of oil price on the optimal speed of ships', *Journal of the Operational Research Society* **33**(11), 1035–1040.

Ronen, D. (1983), 'Cargo ships routing and scheduling: survey of models and problems', *European Journal of Operational Research* **12**(2), 119–126.

Ropke, S. & Cordeau, J.-F. (2009), 'Branch and cut and price for the pickup and delivery problem with time windows', *Transportation Science* **43**(3), 267–286.

Ropke, S. & Pisinger, D. (2006), 'An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows', *Transportation Science* **40**(4), 455–472.

Solomon, M. M. & Desrosiers, J. (1988), 'Survey paper—time window constrained routing and scheduling problems', *Transportation Science* **22**(1), 1–13.

Subramanian, A., Uchoa, E. & Ochi, L. S. (2013), 'A hybrid algorithm for a class of vehicle routing problems', *Computers & Operations Research* **40**(10), 2519–2531.

Toth, P. & Vigo, D. (2003), 'The granular tabu search and its application to the vehicle-routing problem', *INFORMS Journal on Computing* **15**(4), 333–346.

UNCTAD (2017), 'Review of maritime transport'.

Velasco, N., Castagliola, P., Dejax, P., Guéret, C. & Prins, C. (2009), *A memetic algorithm for a pick-up and delivery problem by helicopter*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 173–190.

Vidal, T. (2016), 'Technical note: split algorithm in o(n) for the capacitated vehicle routing problem', *Computers & Operations Research* **69**, 40–47.

Vidal, T., Crainic, T. G., Gendreau, M. & Prins, C. (2013), 'A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows', *Computers & Operations Research* **40**(1), 475 – 489.

Vidal, T., Crainic, T. G., Gendreau, M. & Prins, C. (2014), 'A unified solution framework for multi-attribute vehicle routing problems', *European Journal of Operational Research* **234**(3), 658 – 673.

Vidal, T., Crainic, T. G., Gendreau, M. & Prins, C. (2015*a*), 'Time-window relaxations in vehicle routing heuristics', *Journal of Heuristics* **21**(3), 329–358.

Vidal, T., Crainic, T. G., Gendreau, M. & Prins, C. (2015*b*), 'Timing problems and algorithms: time decisions for sequences of activities', *Networks* **65**(2), 102–128.

Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N. & Rei, W. (2012), 'A hybrid genetic algorithm for multidepot and periodic vehicle routing problems', *Operations Research* **60**(3), 611–624.

Vidal, T., Gribel, D. & Jaillet, P. (2017), 'Separable convex optimization with nested lower and upper constraints', *ArXiv e-prints*.

Vidal, T., Maculan, N., Ochi, L. & Penna, P. (2016), 'Large neighborhoods with implicit customer selection for vehicle routing problems with profits', *Transportation Science* **50**(2), 720–734.

Vilhelmsen, C., Lusby, R. & Larsen, J. (2014), 'Tramp ship routing and scheduling with integrated bunker optimization', *EURO Journal on Transportation and Logistics* **3**(2), 143–175.

Wang, S. & Meng, Q. (2012), 'Sailing speed optimization for container ships in a liner shipping network', *Transportation Research Part E: Logistics and Transportation Review* **48**(3), 701–714.

Wen, M., Ropke, S., Petersen, H., Larsen, R. & Madsen, O. (2016), 'Full-shipload tramp ship routing and scheduling with variable speeds', *Computers & Operations Research* **70**, 1–8.

A
Appendix

Full HGS results on the ITSRSP instances are displayed in Tables A.1 to A.4. The best, average and worst travel cost found over 10 runs is reported in columns “Best”, “Avg” and “Worst”, respectively. Column “T” gives the average CPU time (in minutes).

Full B&P results are displayed in Tables A.5 to A.8. Columns “LB” and “UB” give respectively the lower and upper bounds. Column “T” gives the execution time (in seconds). Column “EN” gives the number of explored nodes, and column “HN” gives the number of hidden nodes.

Full results for the HGS-O and the HGS-J on the ITSRSPSO instances are displayed in Tables A.9 to A.12. Costs over five runs are reported in columns “Best”, “Avg” and “Worst”. Column “T” gives the average CPU time (in minutes).

Table A.1: Full results on Hemmati et al. (2014) for HGS-NO and HGS-SP.

	HGS-NO				**HGS-SP**			
Instance	Best	Avg	Worst	T	Best	Avg	Worst	T
SHORTSEA_MUN_C7_V3_HE_1	1476444	1476444.00	1476444	0.01	1476444	1476444.00	1476444	0.02
SHORTSEA_MUN_C7_V3_HE_2	1134176	1134176.00	1134176	0.02	1134176	1134176.00	1134176	0.02
SHORTSEA_MUN_C7_V3_HE_3	1196466	1196466.00	1196466	0.02	1196466	1196466.00	1196466	0.02
SHORTSEA_MUN_C7_V3_HE_4	1256139	1256139.00	1256139	0.02	1256139	1256139.00	1256139	0.02
SHORTSEA_MUN_C7_V3_HE_5	1160394	1160394.00	1160394	0.01	1160394	1160394.00	1160394	0.02
SHORTSEA_MUN_C10_V3_HE_1	2083965	2083965.00	2083965	0.03	2083965	2083965.00	2083965	0.04
SHORTSEA_MUN_C10_V3_HE_2	2012364	2012364.00	2012364	0.03	2012364	2012364.00	2012364	0.04
SHORTSEA_MUN_C10_V3_HE_3	1986779	1986779.00	1986779	0.03	1986779	1986779.00	1986779	0.03
SHORTSEA_MUN_C10_V3_HE_4	2125461	2125461.00	2125461	0.03	2125461	2125461.00	2125461	0.03
SHORTSEA_MUN_C10_V3_HE_5	2162453	2162453.00	2162453	0.03	2162453	2162453.00	2162453	0.03
SHORTSEA_MUN_C15_V4_HE_1	1959153	1959153.00	1959153	0.06	1959153	1959153.00	1959153	0.07
SHORTSEA_MUN_C15_V4_HE_2	2560004	2560004.00	2560004	0.07	2560004	2560004.00	2560004	0.07
SHORTSEA_MUN_C15_V4_HE_3	2582912	2582912.00	2582912	0.07	2582912	2582912.00	2582912	0.07
SHORTSEA_MUN_C15_V4_HE_4	2265396	2265396.00	2265396	0.08	2265396	2265396.00	2265396	0.08
SHORTSEA_MUN_C15_V4_HE_5	2230861	2230861.00	2230861	0.06	2230861	2230861.00	2230861	0.07
SHORTSEA_MUN_C18_V5_HE_1	2374420	2374420.00	2374420	0.09	2374420	2374420.00	2374420	0.09
SHORTSEA_MUN_C18_V5_HE_2	2987358	2987358.00	2987358	0.09	2987358	2987358.00	2987358	0.10
SHORTSEA_MUN_C18_V5_HE_3	2301308	2301308.00	2301308	0.10	2301308	2301308.00	2301308	0.10
SHORTSEA_MUN_C18_V5_HE_4	2400016	2410332.50	2428601	0.13	2400016	2401507.60	2414932	0.14
SHORTSEA_MUN_C18_V5_HE_5	2813167	2813763.60	2816150	0.13	2813167	2813167.00	2813167	0.13
SHORTSEA_MUN_C22_V6_HE_1	3928483	3928483.00	3928483	0.13	3928483	3928483.00	3928483	0.13
SHORTSEA_MUN_C22_V6_HE_2	3683436	3683436.00	3683436	0.14	3683436	3683436.00	3683436	0.14
SHORTSEA_MUN_C22_V6_HE_3	3264770	3264770.00	3264770	0.13	3264770	3264770.00	3264770	0.14
SHORTSEA_MUN_C22_V6_HE_4	3228262	3228262.00	3228262	0.14	3228262	3228262.00	3228262	0.15
SHORTSEA_MUN_C22_V6_HE_5	3770560	3770560.00	3770560	0.13	3770560	3770560.00	3770560	0.14
SHORTSEA_MUN_C23_V13_HE_1	2276832	2276832.00	2276832	0.14	2276832	2276832.00	2276832	0.15
SHORTSEA_MUN_C23_V13_HE_2	2255469	2261452.50	2305997	0.22	2255469	2255469.00	2255469	0.18
SHORTSEA_MUN_C23_V13_HE_3	2362503	2362503.00	2362503	0.19	2362503	2362503.00	2362503	0.17
SHORTSEA_MUN_C23_V13_HE_4	2250110	2250873.10	2257741	0.23	2250110	2250110.00	2250110	0.20
SHORTSEA_MUN_C23_V13_HE_5	2325941	2325941.00	2325941	0.15	2325941	2325941.00	2325941	0.15
SHORTSEA_MUN_C30_V6_HE_1	4958542	4958542.00	4958542	0.34	4958542	4958542.00	4958542	0.24
SHORTSEA_MUN_C30_V6_HE_2	4549708	4550160.60	4554234	0.34	4549708	4549708.00	4549708	0.26
SHORTSEA_MUN_C30_V6_HE_3	4098111	4098111.00	4098111	0.25	4098111	4098111.00	4098111	0.24
SHORTSEA_MUN_C30_V6_HE_4	4449449	4449521.70	4449812	0.31	4449449	4449449.00	4449449	0.30
SHORTSEA_MUN_C30_V6_HE_5	4528514	4528514.00	4528514	0.23	4528514	4528514.00	4528514	0.22
SHORTSEA_MUN_C35_V7_HE_1	4893734	4920016.80	4957949	0.45	4893734	4894505.40	4897591	0.50
SHORTSEA_MUN_C35_V7_HE_2	4533265	4540178.70	4548401	0.40	4533265	4533265.00	4533265	0.40
SHORTSEA_MUN_C35_V7_HE_3	4433847	4433847.00	4433847	0.41	4433847	4433847.00	4433847	0.32
SHORTSEA_MUN_C35_V7_HE_4	4580935	4599980.00	4629068	0.52	4580935	4580935.00	4580935	0.40
SHORTSEA_MUN_C35_V7_HE_5	5542825	5562666.60	5595294	0.43	5511661	5512815.00	5523201	0.42
SHORTSEA_MUN_C60_V13_HE_1	8150597	8168916.70	8193642	1.84	8133385	8136827.40	8150597	2.12
SHORTSEA_MUN_C60_V13_HE_2	7971685	7991077.90	8008264	1.57	7971476	7971538.70	7971685	1.17
SHORTSEA_MUN_C60_V13_HE_3	7633937	7659749.70	7700655	1.64	7604198	7620215.00	7636891	1.40
SHORTSEA_MUN_C60_V13_HE_4	8508321	8553112.40	8627386	1.54	8505125	8505391.40	8505421	1.39
SHORTSEA_MUN_C60_V13_HE_5	8921762	8965048.40	8996365	1.82	8921750	8921752.20	8921761	1.46
SHORTSEA_MUN_C80_V20_HE_1	10312967	10410325.20	10629254	4.20	10289573	10292640.60	10296807	5.10
SHORTSEA_MUN_C80_V20_HE_2	10289830	10326494.40	10377475	4.70	10240618	10241195.50	10246354	2.24
SHORTSEA_MUN_C80_V20_HE_3	9625648	9666363.10	9739462	6.01	9606530	9606530.00	9606530	2.10
SHORTSEA_MUN_C80_V20_HE_4	11444974	11554480.20	11699999	3.27	11302476	11302995.60	11305074	3.79
SHORTSEA_MUN_C80_V20_HE_5	10915609	10988509.70	11037616	3.77	10862563	10862563.00	10862563	1.72
SHORTSEA_MUN_C100_V30_HE_1	12686059	12782256.60	12894417	6.99	12626988	12626988.00	12626988	3.03
SHORTSEA_MUN_C100_V30_HE_2	12842978	12936285.80	13045150	9.32	12774864	12775433.00	12776760	4.22
SHORTSEA_MUN_C100_V30_HE_3	11973900	12034803.50	12090079	7.48	11935332	11935332.00	11935332	4.05
SHORTSEA_MUN_C100_V30_HE_4	13702161	13799089.70	13966302	8.05	13605352	13605874.90	13610581	3.91
SHORTSEA_MUN_C100_V30_HE_5	13326476	13408557.60	13628878	7.92	13240648	13240648.00	13240648	4.42
SHORTSEA_MUN_C130_V40_HE_1	16419109	16512854.60	16667949	12.52	16316051	16316084.70	16316388	10.84
SHORTSEA_MUN_C130_V40_HE_2	16510343	16568550.80	16806478	13.45	16260579	16260728.30	16262072	9.36
SHORTSEA_MUN_C130_V40_HE_3	15707113	15771278.90	15858671	13.78	15537963	15541382.60	15546512	8.11
SHORTSEA_MUN_C130_V40_HE_4	17159564	17191068.80	17265729	14.23	17011065	17012189.90	17014719	12.88
SHORTSEA_MUN_C130_V40_HE_5	18567598	18660824.00	18740877	12.59	18273893	18275037.10	18281798	9.41

Table A.2: Full results on Hemmati et al. (2014) for HGS-NO and HGS-SP.

	HGS				**HGS-SP**			
Instance	Best	Avg	Worst	T	Best	Avg	Worst	T
SHORTSEA_FUN_C8_V3_HE_1	1391997	1391997.00	1391997	0.01	1391997	1391997.00	1391997	0.01
SHORTSEA_FUN_C8_V3_HE_2	1246273	1246273.00	1246273	0.01	1246273	1246273.00	1246273	0.01
SHORTSEA_FUN_C8_V3_HE_3	1698102	1698102.00	1698102	0.01	1698102	1698102.00	1698102	0.01
SHORTSEA_FUN_C8_V3_HE_4	1777637	1777637.00	1777637	0.01	1777637	1777637.00	1777637	0.01
SHORTSEA_FUN_C8_V3_HE_5	1636788	1636788.00	1636788	0.01	1636788	1636788.00	1636788	0.01
SHORTSEA_FUN_C11_V4_HE_1	1052463	1052463.00	1052463	0.02	1052463	1052463.00	1052463	0.02
SHORTSEA_FUN_C11_V4_HE_2	1067139	1067139.00	1067139	0.02	1067139	1067139.00	1067139	0.02
SHORTSEA_FUN_C11_V4_HE_3	1212388	1212388.00	1212388	0.02	1212388	1212388.00	1212388	0.02
SHORTSEA_FUN_C11_V4_HE_4	1185465	1185465.00	1185465	0.02	1185465	1185465.00	1185465	0.02
SHORTSEA_FUN_C11_V4_HE_5	1310285	1310285.00	1310285	0.01	1310285	1310285.00	1310285	0.02
SHORTSEA_FUN_C13_V5_HE_1	2034184	2034184.00	2034184	0.02	2034184	2034184.00	2034184	0.02
SHORTSEA_FUN_C13_V5_HE_2	2043253	2043253.00	2043253	0.02	2043253	2043253.00	2043253	0.02
SHORTSEA_FUN_C13_V5_HE_3	2378283	2378283.00	2378283	0.02	2378283	2378283.00	2378283	0.02
SHORTSEA_FUN_C13_V5_HE_4	2707215	2707215.00	2707215	0.02	2707215	2707215.00	2707215	0.02
SHORTSEA_FUN_C13_V5_HE_5	3011648	3011648.00	3011648	0.02	3011648	3011648.00	3011648	0.02
SHORTSEA_FUN_C16_V6_HE_1	3577005	3577005.00	3577005	0.03	3577005	3577005.00	3577005	0.03
SHORTSEA_FUN_C16_V6_HE_2	3560203	3560203.00	3560203	0.03	3560203	3560203.00	3560203	0.03
SHORTSEA_FUN_C16_V6_HE_3	4081013	4081013.00	4081013	0.03	4081013	4081013.00	4081013	0.03
SHORTSEA_FUN_C16_V6_HE_4	3667080	3667080.00	3667080	0.03	3667080	3667080.00	3667080	0.03
SHORTSEA_FUN_C16_V6_HE_5	3438493	3438493.00	3438493	0.03	3438493	3438493.00	3438493	0.03
SHORTSEA_FUN_C17_V13_HE_1	2265731	2265731.00	2265731	0.04	2265731	2265731.00	2265731	0.04
SHORTSEA_FUN_C17_V13_HE_2	3154165	3154165.00	3154165	0.04	3154165	3154165.00	3154165	0.05
SHORTSEA_FUN_C17_V13_HE_3	2699378	2699378.00	2699378	0.04	2699378	2699378.00	2699378	0.04
SHORTSEA_FUN_C17_V13_HE_4	2806231	2806231.00	2806231	0.04	2806231	2806231.00	2806231	0.04
SHORTSEA_FUN_C17_V13_HE_5	2910814	2910814.00	2910814	0.04	2910814	2910814.00	2910814	0.04
SHORTSEA_FUN_C20_V6_HE_1	2973381	2973381.00	2973381	0.04	2973381	2973381.00	2973381	0.04
SHORTSEA_FUN_C20_V6_HE_2	3206514	3206514.00	3206514	0.05	3206514	3206514.00	3206514	0.05
SHORTSEA_FUN_C20_V6_HE_3	3197445	3197445.00	3197445	0.04	3197445	3197445.00	3197445	0.04
SHORTSEA_FUN_C20_V6_HE_4	3342130	3342130.00	3342130	0.04	3342130	3342130.00	3342130	0.04
SHORTSEA_FUN_C20_V6_HE_5	3156378	3156378.00	3156378	0.04	3156378	3156378.00	3156378	0.04
SHORTSEA_FUN_C25_V7_HE_1	3833588	3833588.00	3833588	0.06	3833588	3833588.00	3833588	0.07
SHORTSEA_FUN_C25_V7_HE_2	3673666	3673666.00	3673666	0.05	3673666	3673666.00	3673666	0.06
SHORTSEA_FUN_C25_V7_HE_3	4238213	4238213.00	4238213	0.06	4238213	4238213.00	4238213	0.07
SHORTSEA_FUN_C25_V7_HE_4	4260762	4260762.00	4260762	0.06	4260762	4260762.00	4260762	0.07
SHORTSEA_FUN_C25_V7_HE_5	4069693	4069693.00	4069693	0.06	4069693	4069693.00	4069693	0.07
SHORTSEA_FUN_C35_V13_HE_1	2986667	2986667.00	2986667	0.11	2986667	2986667.00	2986667	0.12
SHORTSEA_FUN_C35_V13_HE_2	3002973	3003096.60	3004193	0.23	3002973	3002973.00	3002973	0.20
SHORTSEA_FUN_C35_V13_HE_3	3084339	3084376.10	3084710	0.15	3084339	3084339.00	3084339	0.15
SHORTSEA_FUN_C35_V13_HE_4	3952461	3952461.30	3952462	0.18	3952461	3952461.00	3952461	0.16
SHORTSEA_FUN_C35_V13_HE_5	3293086	3293147.10	3293669	0.15	3293086	3293086.00	3293086	0.16
SHORTSEA_FUN_C50_V20_HE_1	7258266	7261774.80	7274756	0.56	7258266	7258266.00	7258266	0.36
SHORTSEA_FUN_C50_V20_HE_2	7458380	7466266.70	7469069	0.43	7452465	7452465.00	7452465	0.37
SHORTSEA_FUN_C50_V20_HE_3	6929079	6949770.20	6969977	0.48	6922293	6922293.00	6922293	0.38
SHORTSEA_FUN_C50_V20_HE_4	8933847	8945057.60	8958527	0.53	8933846	8933846.30	8933847	0.42
SHORTSEA_FUN_C50_V20_HE_5	7322307	7324635.00	7330502	0.54	7322307	7322307.00	7322307	0.39
SHORTSEA_FUN_C70_V30_HE_1	10064530	10079207.60	10099052	1.38	10051856	10051856.00	10051856	0.84
SHORTSEA_FUN_C70_V30_HE_2	10464802	10481727.60	10506846	1.56	10455468	10455468.00	10455468	0.87
SHORTSEA_FUN_C70_V30_HE_3	10273969	10288859.70	10315136	1.68	10172541	10172998.20	10174065	1.06
SHORTSEA_FUN_C70_V30_HE_4	10864114	10911945.10	10946258	1.15	10854036	10854036.00	10854036	0.84
SHORTSEA_FUN_C70_V30_HE_5	10902911	10926787.70	10951401	1.40	10886838	10886838.00	10886838	1.10
SHORTSEA_FUN_C90_V40_HE_1	13413594	13437538.00	13460969	2.82	13361947	13362867.80	13371155	1.69
SHORTSEA_FUN_C90_V40_HE_2	13856061	13891003.60	13920519	2.96	13828112	13828112.00	13828112	1.53
SHORTSEA_FUN_C90_V40_HE_3	12660456	12695713.00	12739629	4.28	12627125	12627125.20	12627126	1.61
SHORTSEA_FUN_C90_V40_HE_4	14440962	14478099.90	14503497	3.64	14406428	14406428.00	14406428	1.60
SHORTSEA_FUN_C90_V40_HE_5	13619099	13660523.20	13721704	3.14	13560830	13560830.00	13560830	1.95
SHORTSEA_FUN_C100_V50_HE_1	13813700	13821891.70	13832541	4.08	13800823	13800823.00	13800823	2.31
SHORTSEA_FUN_C100_V50_HE_2	14682666	14698970.70	14722835	4.55	14644836	14644836.00	14644836	2.27
SHORTSEA_FUN_C100_V50_HE_3	13152622	13182104.30	13210288	4.46	13135505	13135505.00	13135505	2.20
SHORTSEA_FUN_C100_V50_HE_4	14877190	14894679.10	14917842	5.10	14841840	14841840.00	14841840	2.51
SHORTSEA_FUN_C100_V50_HE_5	14050647	14075129.70	14096989	4.67	14009874	14009877.80	14009889	1.94

Table A.3: Full results on Hemmati et al. (2014) for HGS-NO and HGS-SP.

	HGS				**HGS-SP**			
Instance	Best	Avg	Worst	T	Best	Avg	Worst	T
DEEPSEA_MUN_C7_V3_HE_1	5233464	5233464.00	5233464	0.02	5233464	5233464.00	5233464	0.02
DEEPSEA_MUN_C7_V3_HE_2	6053699	6053699.00	6053699	0.03	6053699	6053699.00	6053699	0.02
DEEPSEA_MUN_C7_V3_HE_3	5888949	5888949.00	5888949	0.02	5888949	5888949.00	5888949	0.02
DEEPSEA_MUN_C7_V3_HE_4	6510656	6510656.00	6510656	0.03	6510656	6510656.00	6510656	0.02
DEEPSEA_MUN_C7_V3_HE_5	7220458	7220458.00	7220458	0.02	7220458	7220458.00	7220458	0.02
DEEPSEA_MUN_C10_V3_HE_1	7986248	7986248.00	7986248	0.03	7986248	7986248.00	7986248	0.03
DEEPSEA_MUN_C10_V3_HE_2	7754484	7754484.00	7754484	0.04	7754484	7754484.00	7754484	0.03
DEEPSEA_MUN_C10_V3_HE_3	9499357	9499357.00	9499357	0.03	9499357	9499357.00	9499357	0.03
DEEPSEA_MUN_C10_V3_HE_4	8617192	8617192.00	8617192	0.03	8617192	8617192.00	8617192	0.03
DEEPSEA_MUN_C10_V3_HE_5	8653992	8653992.00	8653992	0.03	8653992	8653992.00	8653992	0.03
DEEPSEA_MUN_C15_V4_HE_1	13467090	13467090.00	13467090	0.06	13467090	13467090.00	13467090	0.06
DEEPSEA_MUN_C15_V4_HE_2	12457251	12457251.00	12457251	0.08	12457251	12457251.00	12457251	0.07
DEEPSEA_MUN_C15_V4_HE_3	12567396	12567396.00	12567396	0.06	12567396	12567396.00	12567396	0.06
DEEPSEA_MUN_C15_V4_HE_4	11764241	11764241.00	11764241	0.07	11764241	11764241.00	11764241	0.07
DEEPSEA_MUN_C15_V4_HE_5	10833640	10833640.00	10833640	0.06	10833640	10833640.00	10833640	0.06
DEEPSEA_MUN_C18_V5_HE_1	43054055	43054055.00	43054055	0.10	43054055	43054055.00	43054055	0.10
DEEPSEA_MUN_C18_V5_HE_2	25068287	25068287.00	25068287	0.10	25068287	25068287.00	25068287	0.11
DEEPSEA_MUN_C18_V5_HE_3	29211238	29211238.00	29211238	0.09	29211238	29211238.00	29211238	0.09
DEEPSEA_MUN_C18_V5_HE_4	32281904	32281904.00	32281904	0.09	32281904	32281904.00	32281904	0.09
DEEPSEA_MUN_C18_V5_HE_5	40718028	40718028.00	40718028	0.09	40718028	40718028.00	40718028	0.09
DEEPSEA_MUN_C22_V6_HE_1	41176718	41176718.00	41176718	0.12	41176718	41176718.00	41176718	0.12
DEEPSEA_MUN_C22_V6_HE_2	37236363	37236363.00	37236363	0.17	37236363	37236363.00	37236363	0.15
DEEPSEA_MUN_C22_V6_HE_3	38215238	38215238.00	38215238	0.14	38215238	38215238.00	38215238	0.14
DEEPSEA_MUN_C22_V6_HE_4	34129809	34129809.00	34129809	0.15	34129809	34129809.00	34129809	0.16
DEEPSEA_MUN_C22_V6_HE_5	46379332	46379332.00	46379332	0.12	46379332	46379332.00	46379332	0.12
DEEPSEA_MUN_C23_V13_HE_1	41002992	41002992.00	41002992	0.15	41002992	41002992.00	41002992	0.15
DEEPSEA_MUN_C23_V13_HE_2	28014147	28014147.00	28014147	0.15	28014147	28014147.00	28014147	0.15
DEEPSEA_MUN_C23_V13_HE_3	29090422	29090422.00	29090422	0.13	29090422	29090422.00	29090422	0.13
DEEPSEA_MUN_C23_V13_HE_4	33685274	33685439.20	33686926	0.18	33685274	33685274.00	33685274	0.17
DEEPSEA_MUN_C23_V13_HE_5	38664843	38664843.00	38664843	0.14	38664843	38664843.00	38664843	0.14
DEEPSEA_MUN_C30_V6_HE_1	19227093	19227093.20	19227094	0.23	19227093	19227093.00	19227093	0.21
DEEPSEA_MUN_C30_V6_HE_2	16784810	16807970.70	16904604	0.31	16784810	16784810.00	16784810	0.28
DEEPSEA_MUN_C30_V6_HE_3	21298546	21309027.50	21402168	0.29	21183928	21193941.20	21208961	0.28
DEEPSEA_MUN_C30_V6_HE_4	21076728	21076728.00	21076728	0.20	21076728	21076728.00	21076728	0.21
DEEPSEA_MUN_C30_V6_HE_5	24490671	24493822.70	24509193	0.30	24490671	24490671.00	24490671	0.25
DEEPSEA_MUN_C35_V7_HE_1	65359491	65440588.00	65570413	0.34	65082675	65082675.00	65082675	0.37
DEEPSEA_MUN_C35_V7_HE_2	54810586	54867979.50	55016029	0.47	54810586	54810586.00	54810586	0.32
DEEPSEA_MUN_C35_V7_HE_3	56182502	56182502.80	56182510	0.38	56182502	56182502.00	56182502	0.32
DEEPSEA_MUN_C35_V7_HE_4	61354812	61368330.60	61489997	0.42	61354812	61354812.00	61354812	0.39
DEEPSEA_MUN_C35_V7_HE_5	63904705	63904705.00	63904705	0.27	63904705	63904705.00	63904705	0.26
DEEPSEA_MUN_C60_V13_HE_1	80779035	81179047.90	81602589	1.91	80649895	80661548.00	80708160	1.94
DEEPSEA_MUN_C60_V13_HE_2	74881110	75244185.50	75847675	2.28	74881109	74881109.20	74881110	1.11
DEEPSEA_MUN_C60_V13_HE_3	92362334	92657898.60	92870090	1.86	91766747	91766833.90	91767616	1.14
DEEPSEA_MUN_C60_V13_HE_4	90323416	91162742.90	92201850	2.31	89702352	89702352.00	89702352	1.39
DEEPSEA_MUN_C60_V13_HE_5	89109660	89301080.00	89766598	1.67	88486544	88486544.00	88486544	1.24
DEEPSEA_MUN_C80_V20_HE_1	70859172	71341637.90	72265959	4.55	70718084	70738509.40	70922338	2.46
DEEPSEA_MUN_C80_V20_HE_2	74444535	74740704.80	75681332	3.60	73558165	73584102.20	73603212	4.10
DEEPSEA_MUN_C80_V20_HE_3	78615067	78996540.20	79407419	3.46	78250612	78250612.00	78250612	2.08
DEEPSEA_MUN_C80_V20_HE_4	76108732	76586051.40	77653002	3.86	75962439	75993240.20	76061556	3.25
DEEPSEA_MUN_C80_V20_HE_5	74351451	75540439.50	76727758	4.36	74162521	74162522.10	74162524	2.47
DEEPSEA_MUN_C100_V30_HE_1	153939004	154825211.00	157456520	6.00	150481912	150497215.70	150522656	7.41
DEEPSEA_MUN_C100_V30_HE_2	152382679	153649456.70	155340239	9.60	150826322	150837539.30	150902591	7.05
DEEPSEA_MUN_C100_V30_HE_3	152433775	153482921.50	154433184	7.14	151027805	151031384.70	151038978	8.49
DEEPSEA_MUN_C100_V30_HE_4	155586124	158301263.40	160614189	6.54	151193009	151204669.80	151256471	7.31
DEEPSEA_MUN_C100_V30_HE_5	161289248	163809733.00	166560867	7.01	159789021	159790522.60	159800907	6.63
DEEPSEA_MUN_C130_V40_HE_1	238775717	241008794.40	243960181	12.37	232625257	232634270.10	232637186	14.08
DEEPSEA_MUN_C130_V40_HE_2	233033136	235230009.70	237753236	12.55	228058626	228184429.60	228240325	14.39
DEEPSEA_MUN_C130_V40_HE_3	244438374	247157840.70	250071099	12.63	235666249	235742810.90	235949491	13.87
DEEPSEA_MUN_C130_V40_HE_4	225692405	228146599.60	231896479	13.57	220357686	220419662.80	220762578	14.06
DEEPSEA_MUN_C130_V40_HE_5	243557347	245724607.50	247919101	13.35	235381937	235586360.10	235752606	14.91

Table A.4: Full results on Hemmati et al. (2014) for HGS-NO and HGS-SP.

	HGS				**HGS-SP**			
Instance	Best	Avg	Worst	T	Best	Avg	Worst	T
DEEPSEA_FUN_C8_V3_HE_1	9584863	9584863.00	9584863	0.01	9584863	9584863.00	9584863	0.01
DEEPSEA_FUN_C8_V3_HE_2	9369654	9369654.00	9369654	0.01	9369654	9369654.00	9369654	0.01
DEEPSEA_FUN_C8_V3_HE_3	4596681	4596681.00	4596681	0.01	4596681	4596681.00	4596681	0.01
DEEPSEA_FUN_C8_V3_HE_4	6899730	6899730.00	6899730	0.01	6899730	6899730.00	6899730	0.01
DEEPSEA_FUN_C8_V3_HE_5	6815253	6815253.00	6815253	0.01	6815253	6815253.00	6815253	0.01
DEEPSEA_FUN_C11_V4_HE_1	34854819	34854819.00	34854819	0.01	34854819	34854819.00	34854819	0.02
DEEPSEA_FUN_C11_V4_HE_2	25454434	25454434.00	25454434	0.02	25454434	25454434.00	25454434	0.02
DEEPSEA_FUN_C11_V4_HE_3	29627143	29627143.00	29627143	0.02	29627143	29627143.00	29627143	0.02
DEEPSEA_FUN_C11_V4_HE_4	33111680	33111680.00	33111680	0.02	33111680	33111680.00	33111680	0.02
DEEPSEA_FUN_C11_V4_HE_5	28175914	28175914.00	28175914	0.01	28175914	28175914.00	28175914	0.02
DEEPSEA_FUN_C13_V5_HE_1	11629005	11629005.00	11629005	0.02	11629005	11629005.00	11629005	0.02
DEEPSEA_FUN_C13_V5_HE_2	11820655	11820655.00	11820655	0.02	11820655	11820655.00	11820655	0.02
DEEPSEA_FUN_C13_V5_HE_3	9992593	9992593.00	9992593	0.02	9992593	9992593.00	9992593	0.02
DEEPSEA_FUN_C13_V5_HE_4	12819619	12819619.00	12819619	0.02	12819619	12819619.00	12819619	0.02
DEEPSEA_FUN_C13_V5_HE_5	10534892	10534892.00	10534892	0.02	10534892	10534892.00	10534892	0.02
DEEPSEA_FUN_C16_V6_HE_1	51127590	51127590.00	51127590	0.03	51127590	51127590.00	51127590	0.03
DEEPSEA_FUN_C16_V6_HE_2	44342796	44342796.00	44342796	0.03	44342796	44342796.00	44342796	0.03
DEEPSEA_FUN_C16_V6_HE_3	45391842	45391842.00	45391842	0.03	45391842	45391842.00	45391842	0.03
DEEPSEA_FUN_C16_V6_HE_4	39687114	39687114.00	39687114	0.03	39687114	39687114.00	39687114	0.03
DEEPSEA_FUN_C16_V6_HE_5	42855603	42855603.00	42855603	0.03	42855603	42855603.00	42855603	0.04
DEEPSEA_FUN_C17_V13_HE_1	17316720	17316720.00	17316720	0.04	17316720	17316720.00	17316720	0.04
DEEPSEA_FUN_C17_V13_HE_2	12194861	12194861.00	12194861	0.04	12194861	12194861.00	12194861	0.04
DEEPSEA_FUN_C17_V13_HE_3	12091554	12091554.00	12091554	0.04	12091554	12091554.00	12091554	0.04
DEEPSEA_FUN_C17_V13_HE_4	12847653	12847653.00	12847653	0.04	12847653	12847653.00	12847653	0.05
DEEPSEA_FUN_C17_V13_HE_5	13213406	13213406.00	13213406	0.04	13213406	13213406.00	13213406	0.04
DEEPSEA_FUN_C20_V6_HE_1	16406738	16406738.00	16406738	0.04	16406738	16406738.00	16406738	0.04
DEEPSEA_FUN_C20_V6_HE_2	16079401	16079401.00	16079401	0.04	16079401	16079401.00	16079401	0.04
DEEPSEA_FUN_C20_V6_HE_3	17342200	17342200.00	17342200	0.04	17342200	17342200.00	17342200	0.04
DEEPSEA_FUN_C20_V6_HE_4	16529748	16529748.00	16529748	0.04	16529748	16529748.00	16529748	0.04
DEEPSEA_FUN_C20_V6_HE_5	17449378	17449378.00	17449378	0.04	17449378	17449378.00	17449378	0.04
DEEPSEA_FUN_C25_V7_HE_1	22773158	22773158.00	22773158	0.06	22773158	22773158.00	22773158	0.06
DEEPSEA_FUN_C25_V7_HE_2	20206329	20206329.00	20206329	0.07	20206329	20206329.00	20206329	0.07
DEEPSEA_FUN_C25_V7_HE_3	19108952	19108952.00	19108952	0.06	19108952	19108952.00	19108952	0.06
DEEPSEA_FUN_C25_V7_HE_4	22668675	22668675.00	22668675	0.06	22668675	22668675.00	22668675	0.06
DEEPSEA_FUN_C25_V7_HE_5	23036603	23036603.00	23036603	0.07	23036603	23036603.00	23036603	0.07
DEEPSEA_FUN_C35_V13_HE_1	86951609	86951609.60	86951611	0.24	86951609	86951609.00	86951609	0.20
DEEPSEA_FUN_C35_V13_HE_2	83422071	83422071.00	83422071	0.21	83422071	83422071.00	83422071	0.18
DEEPSEA_FUN_C35_V13_HE_3	83898591	83898591.00	83898591	0.17	83898591	83898591.00	83898591	0.17
DEEPSEA_FUN_C35_V13_HE_4	91970481	91970481.00	91970481	0.24	91970481	91970481.00	91970481	0.20
DEEPSEA_FUN_C35_V13_HE_5	91123040	91159811.60	91216286	0.20	91123040	91123040.00	91123040	0.19
DEEPSEA_FUN_C50_V20_HE_1	41313566	41326002.70	41381501	0.66	41310946	41310946.00	41310946	0.39
DEEPSEA_FUN_C50_V20_HE_2	37825509	37888723.20	37947319	0.51	37784994	37784994.00	37784994	0.43
DEEPSEA_FUN_C50_V20_HE_3	39844745	39897627.00	39943733	0.68	39841724	39841724.00	39841724	0.37
DEEPSEA_FUN_C50_V20_HE_4	43941098	43986102.90	44011291	0.45	43941098	43941098.00	43941098	0.38
DEEPSEA_FUN_C50_V20_HE_5	41947437	41984135.90	42143433	0.62	41947437	41947437.00	41947437	0.38
DEEPSEA_FUN_C70_V30_HE_1	142680222	142723400.50	142799012	1.20	142679953	142679953.20	142679954	0.91
DEEPSEA_FUN_C70_V30_HE_2	135342011	135738652.90	136013952	1.21	135031988	135031988.00	135031988	0.99
DEEPSEA_FUN_C70_V30_HE_3	162773501	162905956.30	163268139	1.43	162759203	162759203.00	162759203	0.89
DEEPSEA_FUN_C70_V30_HE_4	155964893	156117951.00	156254078	1.43	155855123	155855123.00	155855123	0.88
DEEPSEA_FUN_C70_V30_HE_5	156704112	156872711.50	157041679	1.66	156557723	156557723.00	156557723	0.85
DEEPSEA_FUN_C90_V40_HE_1	191055926	191192052.80	191435179	2.67	190627186	190628742.10	190642742	1.85
DEEPSEA_FUN_C90_V40_HE_2	190028139	190249888.10	190725101	2.87	189770977	189771678.30	189777990	2.16
DEEPSEA_FUN_C90_V40_HE_3	211265485	211436543.20	211600313	3.35	211038412	211038626.90	211040561	2.10
DEEPSEA_FUN_C90_V40_HE_4	210564220	210704075.00	210868247	3.70	210449287	210449287.00	210449287	1.71
DEEPSEA_FUN_C90_V40_HE_5	198084330	198352390.50	198672087	3.78	197804917	197804917.00	197804917	1.91
DEEPSEA_FUN_C100_V50_HE_1	205965394	206152838.40	206399739	4.74	205826535	205826658.30	205827768	2.66
DEEPSEA_FUN_C100_V50_HE_2	208251869	208472467.40	208737288	4.25	207809147	207809147.00	207809147	3.05
DEEPSEA_FUN_C100_V50_HE_3	217551154	218077707.80	218927606	4.70	217000928	217000928.00	217000928	2.40
DEEPSEA_FUN_C100_V50_HE_4	221017064	221273598.90	221508378	6.22	220879632	220880010.80	220882880	2.51
DEEPSEA_FUN_C100_V50_HE_5	223346039	223606855.80	223876870	3.91	223265017	223265017.10	223265018	2.52

Table A.5: Full results on Hemmati et al. (2014) for B&P.

	Root Node		MIP		B&P				
Instance	LB	T	UB	T	LB	UB	T	EN	HN
SHORTSEA_MUN_C7_V3_HE_1	1476444.0	0.0	1476444	0.0	1476444.0	1476444	0.0	1	0
SHORTSEA_MUN_C7_V3_HE_2	1134176.0	0.0	1134176	0.0	1134176.0	1134176	0.0	1	0
SHORTSEA_MUN_C7_V3_HE_3	1196466.0	0.0	1196466	0.0	1196466.0	1196466	0.0	1	0
SHORTSEA_MUN_C7_V3_HE_4	1256139.0	0.0	1256139	0.0	1256139.0	1256139	0.0	1	0
SHORTSEA_MUN_C7_V3_HE_5	1160394.0	0.0	1160394	0.0	1160394.0	1160394	0.0	1	0
SHORTSEA_MUN_C10_V3_HE_1	2083965.0	0.0	2083965	0.0	2083965.0	2083965	0.0	1	0
SHORTSEA_MUN_C10_V3_HE_2	2012364.0	0.0	2012364	0.0	2012364.0	2012364	0.0	1	0
SHORTSEA_MUN_C10_V3_HE_3	1986779.0	0.0	1986779	0.0	1986779.0	1986779	0.0	1	0
SHORTSEA_MUN_C10_V3_HE_4	2125461.0	0.0	2125461	0.0	2125461.0	2125461	0.0	1	0
SHORTSEA_MUN_C10_V3_HE_5	2162453.0	0.0	2162453	0.0	2162453.0	2162453	0.0	1	0
SHORTSEA_MUN_C15_V4_HE_1	1959153.0	0.0	1959153	0.0	1959153.0	1959153	0.0	1	0
SHORTSEA_MUN_C15_V4_HE_2	2538961.0	0.0	2560004	0.0	2560004.0	2560004	0.1	7	16
SHORTSEA_MUN_C15_V4_HE_3	2484024.5	0.0	2587984	0.0	2582912.0	2582912	0.1	5	6
SHORTSEA_MUN_C15_V4_HE_4	2252946.0	0.0	2265396	0.0	2265396.0	2265396	0.1	3	2
SHORTSEA_MUN_C15_V4_HE_5	2230861.0	0.0	2230861	0.0	2230861.0	2230861	0.0	1	0
SHORTSEA_MUN_C18_V5_HE_1	2329225.5	0.1	2416632	0.1	2374420.0	2374420	0.6	15	100
SHORTSEA_MUN_C18_V5_HE_2	2987358.0	0.1	2987358	0.1	2987358.0	2987358	0.1	1	0
SHORTSEA_MUN_C18_V5_HE_3	2301308.0	0.0	2301308	0.0	2301308.0	2301308	0.0	1	0
SHORTSEA_MUN_C18_V5_HE_4	2400016.0	0.1	2400016	0.1	2400016.0	2400016	0.1	1	0
SHORTSEA_MUN_C18_V5_HE_5	2789598.0	0.1	2816150	0.1	2813167.0	2813167	0.3	9	64
SHORTSEA_MUN_C22_V6_HE_1	3928483.0	0.0	3928483	0.0	3928483.0	3928483	0.0	1	0
SHORTSEA_MUN_C22_V6_HE_2	3487768.7	0.1	3774389	0.1	3683436.0	3683436	0.9	27	46
SHORTSEA_MUN_C22_V6_HE_3	3231622.0	0.1	3308030	0.1	3264770.0	3264770	1.4	11	245
SHORTSEA_MUN_C22_V6_HE_4	3222544.8	0.2	3228262	0.2	3228262.0	3228262	0.3	3	2
SHORTSEA_MUN_C22_V6_HE_5	3763700.8	0.0	3771688	0.0	3770560.0	3770560	0.1	3	2
SHORTSEA_MUN_C23_V13_HE_1	2249158.2	0.2	2293525	0.3	2276832.0	2276832	4.8	37	221
SHORTSEA_MUN_C23_V13_HE_2	2255469.0	0.3	2255469	0.3	2255469.0	2255469	0.3	1	0
SHORTSEA_MUN_C23_V13_HE_3	2362503.0	0.2	2362503	0.2	2362503.0	2362503	0.2	1	0
SHORTSEA_MUN_C23_V13_HE_4	2231516.5	0.9	2253640	0.9	2250110.0	2250110	37.0	109	995
SHORTSEA_MUN_C23_V13_HE_5	2325747.2	0.1	2325941	0.1	2325941.0	2325941	0.2	3	2
SHORTSEA_MUN_C30_V6_HE_1	4958542.0	0.2	4958542	0.2	4958542.0	4958542	0.2	1	0
SHORTSEA_MUN_C30_V6_HE_2	4495744.8	0.3	4549708	0.3	4549708.0	4549708	1.6	13	82
SHORTSEA_MUN_C30_V6_HE_3	4034469.5	0.9	4159662	1.1	4098111.0	4098111	20.5	39	482
SHORTSEA_MUN_C30_V6_HE_4	4410463.2	0.8	4616521	1.2	4449449.0	4449449	8.4	13	370
SHORTSEA_MUN_C30_V6_HE_5	4528514.0	0.6	4528514	0.6	4528514.0	4528514	0.6	1	0
SHORTSEA_MUN_C35_V7_HE_1	4866614.2	0.8	4985551	1.0	4893734.0	4893734	3.3	7	68
SHORTSEA_MUN_C35_V7_HE_2	4460401.9	2.5	4718541	3.5	4533265.0	4533265	90.3	91	2161
SHORTSEA_MUN_C35_V7_HE_3	4430107.9	2.2	4601192	2.7	4433847.0	4433847	5.5	3	70
SHORTSEA_MUN_C35_V7_HE_4	4548437.7	2.5	4693288	2.9	4580935.0	4580935	22.3	19	394
SHORTSEA_MUN_C35_V7_HE_5	5495244.0	0.7	5527520	0.8	5511661.0	5511661	4.1	5	148
SHORTSEA_MUN_C60_V13_HE_1	8099191.6	33.3	8249940	50.8	8133385.0	8133385	1162.6	145	4226
SHORTSEA_MUN_C60_V13_HE_2	7931646.5	38.9	8036756	47.4	7971476.0	7971476	1761.5	237	4247
SHORTSEA_MUN_C60_V13_HE_3	7599147.4	43.8	7733822	62.2	7604198.0	7604198	94.8	3	57
SHORTSEA_MUN_C60_V13_HE_4	8447115.5	19.5	8643183	37.8	8505125.0	8505125	1165.2	223	6336
SHORTSEA_MUN_C60_V13_HE_5	8881967.7	10.3	8973580	12.7	8921750.0	8921750	266.1	57	1593
SHORTSEA_MUN_C80_V20_HE_1	10255116.5	71.2	10326279	74.0	10289573.0	10289573	1279.1	51	1529
SHORTSEA_MUN_C80_V20_HE_2	10211093.0	55.3	10292036	56.4	10240618.0	10240618	259.2	9	207
SHORTSEA_MUN_C80_V20_HE_3	9596762.2	177.4	9627549	178.1	9606530.0	9606530	1282.1	19	522
SHORTSEA_MUN_C80_V20_HE_4	11280736.1	24.7	11330979	25.6	11302476.0	11302476	173.4	15	31
SHORTSEA_MUN_C80_V20_HE_5	10858161.5	23.2	10882782	23.6	10862563.0	10862563	54.1	5	59
SHORTSEA_MUN_C100_V30_HE_1	12612920.4	128.2	12641227	129.8	12625185.1	12641227	3600.0	107	2710
SHORTSEA_MUN_C100_V30_HE_2	12744751.8	115.9	12784951	116.8	12773692.8	12776760	3600.0	103	2987
SHORTSEA_MUN_C100_V30_HE_3	11910321.0	345.2	11946818	347.6	11928269.6	11946818	3600.0	41	1125
SHORTSEA_MUN_C100_V30_HE_4	13596484.5	107.5	13607851	107.8	13605352.0	13605352	1145.1	35	440
SHORTSEA_MUN_C100_V30_HE_5	13225091.3	52.1	13251288	52.9	13240648.0	13240648	1263.7	55	930
SHORTSEA_MUN_C130_V40_HE_1	16286675.8	369.3	16342248	456.0	16308671.9	16342248	3600.0	26	759
SHORTSEA_MUN_C130_V40_HE_2	16243044.1	1324.0	16288691	1341.1	16246161.3	16288691	3600.0	8	73
SHORTSEA_MUN_C130_V40_HE_3	15510595.3	793.5	15545580	795.4	15520539.8	15545580	3600.0	13	183
SHORTSEA_MUN_C130_V40_HE_4	16998857.1	235.1	17020967	248.5	17007914.4	17020967	3600.0	55	1249
SHORTSEA_MUN_C130_V40_HE_5	18251952.7	428.0	18275526	429.9	18264649.5	18275526	3600.0	34	483

Table A.6: Full results on Hemmati et al. (2014) for B&P.

	Root Node		MIP		B&P				
Instance	LB	T	UB	T	LB	UB	T	EN	HN
SHORTSEA_FUN_C8_V3_HE_1	1391997.0	0.0	1391997	0.0	1391997.0	1391997	0.0	1	0
SHORTSEA_FUN_C8_V3_HE_2	1246273.0	0.0	1246273	0.0	1246273.0	1246273	0.0	1	0
SHORTSEA_FUN_C8_V3_HE_3	1698102.0	0.0	1698102	0.0	1698102.0	1698102	0.0	1	0
SHORTSEA_FUN_C8_V3_HE_4	1777637.0	0.0	1777637	0.0	1777637.0	1777637	0.0	1	0
SHORTSEA_FUN_C8_V3_HE_5	1636788.0	0.0	1636788	0.0	1636788.0	1636788	0.0	1	0
SHORTSEA_FUN_C11_V4_HE_1	1052463.0	0.0	1052463	0.0	1052463.0	1052463	0.0	1	0
SHORTSEA_FUN_C11_V4_HE_2	1067139.0	0.0	1067139	0.0	1067139.0	1067139	0.0	1	0
SHORTSEA_FUN_C11_V4_HE_3	1212388.0	0.0	1212388	0.0	1212388.0	1212388	0.0	1	0
SHORTSEA_FUN_C11_V4_HE_4	1185465.0	0.0	1185465	0.0	1185465.0	1185465	0.0	1	0
SHORTSEA_FUN_C11_V4_HE_5	1310285.0	0.0	1310285	0.0	1310285.0	1310285	0.0	1	0
SHORTSEA_FUN_C13_V5_HE_1	2034184.0	0.0	2034184	0.0	2034184.0	2034184	0.0	1	0
SHORTSEA_FUN_C13_V5_HE_2	2043253.0	0.0	2043253	0.0	2043253.0	2043253	0.0	1	0
SHORTSEA_FUN_C13_V5_HE_3	2378283.0	0.0	2378283	0.0	2378283.0	2378283	0.0	1	0
SHORTSEA_FUN_C13_V5_HE_4	2707215.0	0.0	2707215	0.0	2707215.0	2707215	0.0	1	0
SHORTSEA_FUN_C13_V5_HE_5	3011648.0	0.0	3011648	0.0	3011648.0	3011648	0.0	1	0
SHORTSEA_FUN_C16_V6_HE_1	3577005.0	0.0	3577005	0.0	3577005.0	3577005	0.0	1	0
SHORTSEA_FUN_C16_V6_HE_2	3560203.0	0.0	3560203	0.0	3560203.0	3560203	0.0	1	0
SHORTSEA_FUN_C16_V6_HE_3	4081013.0	0.0	4081013	0.0	4081013.0	4081013	0.0	1	0
SHORTSEA_FUN_C16_V6_HE_4	3667080.0	0.0	3667080	0.0	3667080.0	3667080	0.0	1	0
SHORTSEA_FUN_C16_V6_HE_5	3438493.0	0.0	3438493	0.0	3438493.0	3438493	0.0	1	0
SHORTSEA_FUN_C17_V13_HE_1	2265329.0	0.0	2265731	0.0	2265731.0	2265731	0.0	3	2
SHORTSEA_FUN_C17_V13_HE_2	3154165.0	0.0	3154165	0.0	3154165.0	3154165	0.0	1	0
SHORTSEA_FUN_C17_V13_HE_3	2697988.7	0.0	2699378	0.0	2699378.0	2699378	0.0	3	2
SHORTSEA_FUN_C17_V13_HE_4	2806231.0	0.0	2806231	0.0	2806231.0	2806231	0.0	1	0
SHORTSEA_FUN_C17_V13_HE_5	2910814.0	0.0	2910814	0.0	2910814.0	2910814	0.0	1	0
SHORTSEA_FUN_C20_V6_HE_1	2973381.0	0.0	2973381	0.0	2973381.0	2973381	0.0	1	0
SHORTSEA_FUN_C20_V6_HE_2	3206514.0	0.0	3206514	0.0	3206514.0	3206514	0.0	1	0
SHORTSEA_FUN_C20_V6_HE_3	3192228.6	0.0	3200652	0.0	3197445.0	3197445	0.1	3	30
SHORTSEA_FUN_C20_V6_HE_4	3342117.5	0.0	3342130	0.0	3342130.0	3342130	0.0	3	4
SHORTSEA_FUN_C20_V6_HE_5	3156378.0	0.0	3156378	0.0	3156378.0	3156378	0.0	1	0
SHORTSEA_FUN_C25_V7_HE_1	3833588.0	0.0	3833588	0.0	3833588.0	3833588	0.0	1	0
SHORTSEA_FUN_C25_V7_HE_2	3673666.0	0.0	3673666	0.0	3673666.0	3673666	0.0	1	0
SHORTSEA_FUN_C25_V7_HE_3	4237512.0	0.0	4250780	0.0	4238213.0	4238213	0.1	3	34
SHORTSEA_FUN_C25_V7_HE_4	4260762.0	0.0	4260762	0.0	4260762.0	4260762	0.0	1	0
SHORTSEA_FUN_C25_V7_HE_5	4069693.0	0.0	4069693	0.0	4069693.0	4069693	0.0	1	0
SHORTSEA_FUN_C35_V13_HE_1	2986667.0	0.1	2986667	0.1	2986667.0	2986667	0.1	1	0
SHORTSEA_FUN_C35_V13_HE_2	3002595.7	0.1	3002974	0.1	3002973.0	3002973	0.2	3	2
SHORTSEA_FUN_C35_V13_HE_3	3084339.0	0.1	3084339	0.1	3084339.0	3084339	0.1	1	0
SHORTSEA_FUN_C35_V13_HE_4	3952461.0	0.1	3952461	0.1	3952461.0	3952461	0.1	1	0
SHORTSEA_FUN_C35_V13_HE_5	3293086.0	0.1	3293086	0.1	3293086.0	3293086	0.1	1	0
SHORTSEA_FUN_C50_V20_HE_1	7258266.0	0.2	7258266	0.2	7258266.0	7258266	0.2	1	0
SHORTSEA_FUN_C50_V20_HE_2	7452465.0	0.2	7452465	0.2	7452465.0	7452465	0.2	1	0
SHORTSEA_FUN_C50_V20_HE_3	6922293.0	0.2	6922293	0.2	6922293.0	6922293	0.2	1	0
SHORTSEA_FUN_C50_V20_HE_4	8933721.0	0.2	8933847	0.2	8933846.0	8933846	0.8	3	41
SHORTSEA_FUN_C50_V20_HE_5	7312328.5	0.2	7322307	0.2	7322307.0	7322307	0.5	3	4
SHORTSEA_FUN_C70_V30_HE_1	10051856.0	1.0	10051856	1.0	10051856.0	10051856	1.0	1	0
SHORTSEA_FUN_C70_V30_HE_2	10455468.0	0.7	10455468	0.7	10455468.0	10455468	0.7	1	0
SHORTSEA_FUN_C70_V30_HE_3	10172541.0	0.6	10172541	0.7	10172541.0	10172541	0.7	1	0
SHORTSEA_FUN_C70_V30_HE_4	10851222.8	0.5	10854037	0.6	10854036.0	10854036	4.7	11	24
SHORTSEA_FUN_C70_V30_HE_5	10822672.0	0.6	10888089	0.6	10886838.0	10886838	3.2	13	12
SHORTSEA_FUN_C90_V40_HE_1	13361849.5	1.9	13362182	2.0	13361947.0	13361947	51.2	11	196
SHORTSEA_FUN_C90_V40_HE_2	13828112.0	1.4	13828112	1.4	13828112.0	13828112	1.4	1	0
SHORTSEA_FUN_C90_V40_HE_3	12626910.5	2.0	12627125	2.1	12627125.0	12627125	3.6	3	2
SHORTSEA_FUN_C90_V40_HE_4	14406428.0	1.6	14406428	1.6	14406428.0	14406428	1.6	1	0
SHORTSEA_FUN_C90_V40_HE_5	13559110.5	1.8	13564406	2.1	13560830.0	13560830	8.5	3	61
SHORTSEA_FUN_C100_V50_HE_1	13800823.0	1.8	13800823	1.8	13800823.0	13800823	1.8	1	0
SHORTSEA_FUN_C100_V50_HE_2	14639802.3	1.9	14645667	2.1	14644836.0	14644836	90.6	27	331
SHORTSEA_FUN_C100_V50_HE_3	13135505.0	2.6	13135505	2.6	13135505.0	13135505	2.6	1	0
SHORTSEA_FUN_C100_V50_HE_4	14839249.5	2.7	14841841	3.0	14841840.0	14841840	192.8	55	669
SHORTSEA_FUN_C100_V50_HE_5	14009079.2	2.0	14009874	2.1	14009874.0	14009874	18.1	15	22

Table A.7: Full results on Hemmati et al. (2014) for B&P.

	Root Node		MIP		Branch-and-Price				
Instance	LB	T	UB	T	LB	UB	T	EN	HN
DEEPSEA_MUN_C7_V3_HE_1	4950070.5	0.0	5233464	0.0	5233464.0	5233464	0.0	3	2
DEEPSEA_MUN_C7_V3_HE_2	6053699.0	0.0	6053699	0.0	6053699.0	6053699	0.0	1	0
DEEPSEA_MUN_C7_V3_HE_3	5888949.0	0.0	5888949	0.0	5888949.0	5888949	0.0	1	0
DEEPSEA_MUN_C7_V3_HE_4	6510656.0	0.0	6510656	0.0	6510656.0	6510656	0.0	1	0
DEEPSEA_MUN_C7_V3_HE_5	7220458.0	0.0	7220458	0.0	7220458.0	7220458	0.0	1	0
DEEPSEA_MUN_C10_V3_HE_1	7986248.0	0.0	7986248	0.0	7986248.0	7986248	0.0	1	0
DEEPSEA_MUN_C10_V3_HE_2	7754484.0	0.0	7754484	0.0	7754484.0	7754484	0.0	1	0
DEEPSEA_MUN_C10_V3_HE_3	9295578.0	0.0	9499357	0.0	9499357.0	9499357	0.0	3	4
DEEPSEA_MUN_C10_V3_HE_4	8602011.0	0.0	8617192	0.0	8617192.0	8617192	0.0	5	4
DEEPSEA_MUN_C10_V3_HE_5	8070404.0	0.0	9702181	0.0	8653992.0	8653992	0.0	5	15
DEEPSEA_MUN_C15_V4_HE_1	12812979.5	0.0	13467090	0.0	13467090.0	13467090	0.0	5	9
DEEPSEA_MUN_C15_V4_HE_2	12457251.0	0.0	12457251	0.0	12457251.0	12457251	0.0	1	0
DEEPSEA_MUN_C15_V4_HE_3	12567396.0	0.0	12567396	0.0	12567396.0	12567396	0.0	1	0
DEEPSEA_MUN_C15_V4_HE_4	11764241.0	0.0	11764241	0.0	11764241.0	11764241	0.0	1	0
DEEPSEA_MUN_C15_V4_HE_5	10833640.0	0.0	10833640	0.0	10833640.0	10833640	0.0	1	0
DEEPSEA_MUN_C18_V5_HE_1	42480419.0	0.0	43054055	0.0	43054055.0	43054055	0.0	3	2
DEEPSEA_MUN_C18_V5_HE_2	24770447.0	0.1	25528180	0.1	25068287.0	25068287	0.7	7	56
DEEPSEA_MUN_C18_V5_HE_3	29211238.0	0.0	29211238	0.0	29211238.0	29211238	0.0	1	0
DEEPSEA_MUN_C18_V5_HE_4	32281904.0	0.0	32281904	0.0	32281904.0	32281904	0.0	1	0
DEEPSEA_MUN_C18_V5_HE_5	40718028.0	0.0	40718028	0.0	40718028.0	40718028	0.0	1	0
DEEPSEA_MUN_C22_V6_HE_1	41176718.0	0.0	41176718	0.0	41176718.0	41176718	0.0	1	0
DEEPSEA_MUN_C22_V6_HE_2	37236363.0	0.0	37236363	0.0	37236363.0	37236363	0.0	1	0
DEEPSEA_MUN_C22_V6_HE_3	36724417.0	0.0	38215238	0.0	38215238.0	38215238	0.1	9	12
DEEPSEA_MUN_C22_V6_HE_4	33070380.2	0.1	34364524	0.1	34129809.0	34129809	0.3	9	84
DEEPSEA_MUN_C22_V6_HE_5	46379332.0	0.0	46379332	0.0	46379332.0	46379332	0.0	1	0
DEEPSEA_MUN_C23_V13_HE_1	41002992.0	0.0	41002992	0.0	41002992.0	41002992	0.0	1	0
DEEPSEA_MUN_C23_V13_HE_2	27814256.5	0.1	28014147	0.1	28014147.0	28014147	0.1	3	2
DEEPSEA_MUN_C23_V13_HE_3	29090422.0	0.0	29090422	0.0	29090422.0	29090422	0.0	1	0
DEEPSEA_MUN_C23_V13_HE_4	33471039.5	0.0	33685274	0.0	33685274.0	33685274	0.1	5	4
DEEPSEA_MUN_C23_V13_HE_5	38664843.0	0.0	38664843	0.0	38664843.0	38664843	0.0	1	0
DEEPSEA_MUN_C30_V6_HE_1	19227093.0	0.0	19227093	0.0	19227093.0	19227093	0.0	1	0
DEEPSEA_MUN_C30_V6_HE_2	15950873.6	0.2	17642204	0.3	16784810.0	16784810	6.2	37	344
DEEPSEA_MUN_C30_V6_HE_3	20917898.3	0.1	21431487	0.2	21183928.0	21183928	0.7	13	12
DEEPSEA_MUN_C30_V6_HE_4	21076728.0	0.2	21076728	0.2	21076728.0	21076728	0.2	1	0
DEEPSEA_MUN_C30_V6_HE_5	24177325.6	0.1	24513492	0.1	24490671.0	24490671	2.1	39	348
DEEPSEA_MUN_C35_V7_HE_1	64451358.0	0.1	65119365	0.1	65082675.0	65082675	1.0	13	86
DEEPSEA_MUN_C35_V7_HE_2	53980309.5	1.0	55025454	1.0	54810586.0	54810586	6.5	13	177
DEEPSEA_MUN_C35_V7_HE_3	55814152.0	0.2	56182502	0.2	56182502.0	56182502	1.5	9	80
DEEPSEA_MUN_C35_V7_HE_4	59887019.1	0.2	62723359	0.4	61354812.0	61354812	1.6	15	50
DEEPSEA_MUN_C35_V7_HE_5	63904705.0	0.1	63904705	0.1	63904705.0	63904705	0.1	1	0
DEEPSEA_MUN_C60_V13_HE_1	79933869.8	18.5	81972624	43.5	80649895.0	80649895	984.2	157	4174
DEEPSEA_MUN_C60_V13_HE_2	73811018.5	6.3	78034376	7.4	74881109.0	74881109	138.0	47	1221
DEEPSEA_MUN_C60_V13_HE_3	91718990.0	1.5	91766747	1.6	91766747.0	91766747	2.5	3	5
DEEPSEA_MUN_C60_V13_HE_4	89702352.0	3.2	89702352	3.2	89702352.0	89702352	3.2	1	0
DEEPSEA_MUN_C60_V13_HE_5	88056480.1	3.8	89071597	4.1	88486544.0	88486544	25.4	9	257
DEEPSEA_MUN_C80_V20_HE_1	70345242.4	14.8	71448738	25.9	70718084.0	70718084	354.8	53	918
DEEPSEA_MUN_C80_V20_HE_2	73068858.7	54.3	73921496	60.3	73463298.1	73601236	3600.0	247	6333
DEEPSEA_MUN_C80_V20_HE_3	78104852.0	8.6	78268464	8.6	78250612.0	78250612	30.3	5	71
DEEPSEA_MUN_C80_V20_HE_4	75749235.0	14.7	76486358	20.9	75962439.0	75962439	416.6	77	1118
DEEPSEA_MUN_C80_V20_HE_5	73808271.6	11.3	75315334	13.1	74162521.0	74162521	127.0	27	464
DEEPSEA_MUN_C100_V30_HE_1	150197055.9	29.7	150638897	39.1	150451783.0	150638897	3600.0	277	6775
DEEPSEA_MUN_C100_V30_HE_2	150350415.5	25.4	150862914	25.6	150826322.0	150826322	697.3	79	935
DEEPSEA_MUN_C100_V30_HE_3	149804321.2	23.5	151718112	65.3	150837352.8	151718112	3600.0	341	8198
DEEPSEA_MUN_C100_V30_HE_4	150062944.5	46.7	151890429	55.1	150404783.5	151890429	3600.0	223	4821
DEEPSEA_MUN_C100_V30_HE_5	158977634.6	40.5	160713609	42.2	159789021.0	159789021	1392.4	97	1954
DEEPSEA_MUN_C130_V40_HE_1	232233375.9	200.2	232641884	205.1	232324965.2	232641884	3600.0	52	924
DEEPSEA_MUN_C130_V40_HE_2	226590319.5	215.8	229307168	224.9	226683287.2	229307168	3600.0	69	683
DEEPSEA_MUN_C130_V40_HE_3	234147177.2	170.6	235705442	174.4	235477285.1	235705442	3600.0	60	1059
DEEPSEA_MUN_C130_V40_HE_4	219926810.8	270.6	220540075	277.7	220165826.8	220540075	3600.0	43	752
DEEPSEA_MUN_C130_V40_HE_5	234436256.7	205.6	235661384	209.6	234727321.5	235661384	3600.0	74	590

Table A.8: Full results on Hemmati et al. (2014) for B&P.

	Root Node		MIP		B&P				
Instance	LB	T	UB	T	LB	UB	T	EN	HN
DEEPSEA_FUN_C8_V3_HE_1	9584863.0	0.0	9584863	0.0	9584863.0	9584863	0.0	1	0
DEEPSEA_FUN_C8_V3_HE_2	9369654.0	0.0	9369654	0.0	9369654.0	9369654	0.0	1	0
DEEPSEA_FUN_C8_V3_HE_3	4596681.0	0.0	4596681	0.0	4596681.0	4596681	0.0	1	0
DEEPSEA_FUN_C8_V3_HE_4	6899730.0	0.0	6899730	0.0	6899730.0	6899730	0.0	1	0
DEEPSEA_FUN_C8_V3_HE_5	6815253.0	0.0	6815253	0.0	6815253.0	6815253	0.0	1	0
DEEPSEA_FUN_C11_V4_HE_1	34854819.0	0.0	34854819	0.0	34854819.0	34854819	0.0	1	0
DEEPSEA_FUN_C11_V4_HE_2	25454434.0	0.0	25454434	0.0	25454434.0	25454434	0.0	1	0
DEEPSEA_FUN_C11_V4_HE_3	29627143.0	0.0	29627143	0.0	29627143.0	29627143	0.0	1	0
DEEPSEA_FUN_C11_V4_HE_4	33111680.0	0.0	33111680	0.0	33111680.0	33111680	0.0	1	0
DEEPSEA_FUN_C11_V4_HE_5	28175914.0	0.0	28175914	0.0	28175914.0	28175914	0.0	1	0
DEEPSEA_FUN_C13_V5_HE_1	11629005.0	0.0	11629005	0.0	11629005.0	11629005	0.0	1	0
DEEPSEA_FUN_C13_V5_HE_2	11820655.0	0.0	11820655	0.0	11820655.0	11820655	0.0	1	0
DEEPSEA_FUN_C13_V5_HE_3	9992593.0	0.0	9992593	0.0	9992593.0	9992593	0.0	1	0
DEEPSEA_FUN_C13_V5_HE_4	12819619.0	0.0	12819619	0.0	12819619.0	12819619	0.0	1	0
DEEPSEA_FUN_C13_V5_HE_5	10534892.0	0.0	10534892	0.0	10534892.0	10534892	0.0	1	0
DEEPSEA_FUN_C16_V6_HE_1	51091999.5	0.0	51127590	0.0	51127590.0	51127590	0.0	3	2
DEEPSEA_FUN_C16_V6_HE_2	44342796.0	0.0	44342796	0.0	44342796.0	44342796	0.0	1	0
DEEPSEA_FUN_C16_V6_HE_3	45391842.0	0.0	45391842	0.0	45391842.0	45391842	0.0	1	0
DEEPSEA_FUN_C16_V6_HE_4	39687114.0	0.0	39687114	0.0	39687114.0	39687114	0.0	1	0
DEEPSEA_FUN_C16_V6_HE_5	42855603.0	0.0	42855603	0.0	42855603.0	42855603	0.0	1	0
DEEPSEA_FUN_C17_V13_HE_1	17316720.0	0.0	17316720	0.0	17316720.0	17316720	0.0	1	0
DEEPSEA_FUN_C17_V13_HE_2	12180956.5	0.0	12194861	0.0	12194861.0	12194861	0.0	3	4
DEEPSEA_FUN_C17_V13_HE_3	12091554.0	0.0	12091554	0.0	12091554.0	12091554	0.0	1	0
DEEPSEA_FUN_C17_V13_HE_4	12847653.0	0.0	12847653	0.0	12847653.0	12847653	0.0	1	0
DEEPSEA_FUN_C17_V13_HE_5	13213406.0	0.0	13213406	0.0	13213406.0	13213406	0.0	1	0
DEEPSEA_FUN_C20_V6_HE_1	16315682.5	0.0	16406738	0.0	16406738.0	16406738	0.0	5	6
DEEPSEA_FUN_C20_V6_HE_2	16079401.0	0.0	16079401	0.0	16079401.0	16079401	0.0	1	0
DEEPSEA_FUN_C20_V6_HE_3	17342200.0	0.0	17342200	0.0	17342200.0	17342200	0.0	1	0
DEEPSEA_FUN_C20_V6_HE_4	16529748.0	0.0	16529748	0.0	16529748.0	16529748	0.0	1	0
DEEPSEA_FUN_C20_V6_HE_5	17449378.0	0.0	17449379	0.0	17449378.0	17449378	0.0	3	16
DEEPSEA_FUN_C25_V7_HE_1	22773158.0	0.0	22773158	0.0	22773158.0	22773158	0.0	1	0
DEEPSEA_FUN_C25_V7_HE_2	20206329.0	0.0	20206329	0.0	20206329.0	20206329	0.0	1	0
DEEPSEA_FUN_C25_V7_HE_3	19108952.0	0.0	19108952	0.0	19108952.0	19108952	0.0	1	0
DEEPSEA_FUN_C25_V7_HE_4	22668675.0	0.0	22668675	0.0	22668675.0	22668675	0.0	1	0
DEEPSEA_FUN_C25_V7_HE_5	23036603.0	0.0	23036603	0.0	23036603.0	23036603	0.0	1	0
DEEPSEA_FUN_C35_V13_HE_1	86951609.0	0.1	86951609	0.1	86951609.0	86951609	0.1	1	0
DEEPSEA_FUN_C35_V13_HE_2	83422071.0	0.1	83422071	0.1	83422071.0	83422071	0.1	1	0
DEEPSEA_FUN_C35_V13_HE_3	83898591.0	0.1	83898591	0.1	83898591.0	83898591	0.1	1	0
DEEPSEA_FUN_C35_V13_HE_4	91970481.0	0.1	91970481	0.1	91970481.0	91970481	0.1	1	0
DEEPSEA_FUN_C35_V13_HE_5	91003609.7	0.1	91123040	0.1	91123040.0	91123040	0.6	13	30
DEEPSEA_FUN_C50_V20_HE_1	41310946.0	0.1	41310946	0.1	41310946.0	41310946	0.1	1	0
DEEPSEA_FUN_C50_V20_HE_2	37784994.0	0.1	37784994	0.1	37784994.0	37784994	0.1	1	0
DEEPSEA_FUN_C50_V20_HE_3	39841724.0	0.1	39841724	0.1	39841724.0	39841724	0.1	1	0
DEEPSEA_FUN_C50_V20_HE_4	43941098.0	0.1	43941098	0.1	43941098.0	43941098	0.1	1	0
DEEPSEA_FUN_C50_V20_HE_5	41947437.0	0.1	41947437	0.1	41947437.0	41947437	0.1	1	0
DEEPSEA_FUN_C70_V30_HE_1	142679953.0	0.4	142679953	0.4	142679953.0	142679953	0.4	1	0
DEEPSEA_FUN_C70_V30_HE_2	135008299.5	0.5	135031990	0.6	135031988.0	135031988	1.8	3	35
DEEPSEA_FUN_C70_V30_HE_3	162759203.0	0.3	162759203	0.3	162759203.0	162759203	0.3	1	0
DEEPSEA_FUN_C70_V30_HE_4	155855123.0	0.3	155855123	0.3	155855123.0	155855123	0.3	1	0
DEEPSEA_FUN_C70_V30_HE_5	156557723.0	0.3	156557723	0.3	156557723.0	156557723	0.3	1	0
DEEPSEA_FUN_C90_V40_HE_1	190624194.5	1.0	190629225	1.1	190627186.0	190627186	23.9	9	164
DEEPSEA_FUN_C90_V40_HE_2	189770977.0	0.9	189770977	0.9	189770977.0	189770977	0.9	1	0
DEEPSEA_FUN_C90_V40_HE_3	211038412.0	0.7	211038412	0.8	211038412.0	211038412	0.8	1	0
DEEPSEA_FUN_C90_V40_HE_4	210449287.0	0.8	210449287	0.8	210449287.0	210449287	0.8	1	0
DEEPSEA_FUN_C90_V40_HE_5	197797793.0	0.9	197804918	0.9	197804917.0	197804917	4.2	3	45
DEEPSEA_FUN_C100_V50_HE_1	205824545.5	1.0	205826535	1.1	205826535.0	205826535	2.1	3	2
DEEPSEA_FUN_C100_V50_HE_2	207809147.0	1.3	207809147	1.4	207809147.0	207809147	1.4	1	0
DEEPSEA_FUN_C100_V50_HE_3	217000928.0	1.0	217000928	1.0	217000928.0	217000928	1.0	1	0
DEEPSEA_FUN_C100_V50_HE_4	220879530.5	1.1	220879632	1.2	220879632.0	220879632	2.4	3	2
DEEPSEA_FUN_C100_V50_HE_5	223265017.0	0.9	223265017	0.9	223265017.0	223265017	0.9	1	0

Table A.9: Full results on ITSRSPSO SS_MUN instances.

	HGS-O				HGS-J			
Instance	Best	Avg	Worst	T	Best	Avg	Worst	T
SHORTSEA_MUN_C7_V3_HE_1	1166381.49	1167132.18	1168258.22	0.02	1166381.49	1166381.49	1166381.49	0.16
SHORTSEA_MUN_C7_V3_HE_2	815775.78	818829.30	819592.68	0.02	782788.27	782788.27	782788.27	0.38
SHORTSEA_MUN_C7_V3_HE_3	892411.55	892411.55	892411.55	0.02	892411.55	892411.55	892411.55	0.35
SHORTSEA_MUN_C7_V3_HE_4	770921.80	773398.60	775049.80	0.02	750081.73	750081.73	750081.73	0.36
SHORTSEA_MUN_C7_V3_HE_5	898921.62	898921.62	898921.62	0.02	883520.58	883520.58	883520.58	0.28
SHORTSEA_MUN_C10_V3_HE_1	1592630.53	1593429.41	1596624.91	0.03	1445963.78	1445963.78	1445963.78	0.71
SHORTSEA_MUN_C10_V3_HE_2	1426733.17	1426733.17	1426733.17	0.03	1373919.28	1373919.28	1373919.28	1.03
SHORTSEA_MUN_C10_V3_HE_3	1512591.38	1512591.38	1512591.38	0.04	1496838.85	1496838.85	1496838.85	1.02
SHORTSEA_MUN_C10_V3_HE_4	1613472.87	1613472.87	1613472.87	0.04	1597462.74	1597462.74	1597462.74	1.21
SHORTSEA_MUN_C10_V3_HE_5	1399559.10	1401354.28	1404047.04	0.04	1399559.10	1399559.10	1399559.10	0.75
SHORTSEA_MUN_C15_V4_HE_1	1211743.73	1211753.01	1211790.12	0.07	1199770.91	1199770.91	1199770.91	2.99
SHORTSEA_MUN_C15_V4_HE_2	1927069.71	1937319.74	1943638.17	0.08	1867671.50	1867671.50	1867671.50	3.50
SHORTSEA_MUN_C15_V4_HE_3	1926526.92	1945767.56	1950577.72	0.08	1777836.83	1777836.83	1777836.83	4.92
SHORTSEA_MUN_C15_V4_HE_4	1753962.83	1769996.24	1780685.18	0.09	1446644.28	1446644.28	1446644.28	3.52
SHORTSEA_MUN_C15_V4_HE_5	1704484.62	1706603.90	1709782.82	0.07	1632981.62	1632981.62	1632981.62	3.28
SHORTSEA_MUN_C18_V5_HE_1	1740636.01	1740636.01	1740636.01	0.10	1644116.25	1644116.25	1644116.25	4.27
SHORTSEA_MUN_C18_V5_HE_2	2502280.63	2507044.65	2508235.65	0.12	2339298.58	2339298.58	2339298.58	5.90
SHORTSEA_MUN_C18_V5_HE_3	1767685.42	1770144.47	1779674.32	0.12	1663189.32	1663189.32	1663189.32	6.57
SHORTSEA_MUN_C18_V5_HE_4	1833155.63	1843680.02	1846642.87	0.15	1724844.09	1724844.09	1724844.09	5.65
SHORTSEA_MUN_C18_V5_HE_5	2202180.97	2202180.97	2202180.97	0.12	1845590.32	1845590.32	1845590.32	5.08
SHORTSEA_MUN_C22_V6_HE_1	2483182.85	2483182.85	2483182.85	0.15	2378854.16	2378854.16	2378854.16	6.05
SHORTSEA_MUN_C22_V6_HE_2	2631639.98	2631686.14	2631697.68	0.17	2198359.19	2198359.19	2198359.19	7.16
SHORTSEA_MUN_C22_V6_HE_3	2250128.80	2250128.80	2250128.80	0.18	2041332.32	2041332.32	2041332.32	10.18
SHORTSEA_MUN_C22_V6_HE_4	2181167.68	2181167.68	2181167.68	0.17	2053885.58	2053885.58	2053885.58	10.41
SHORTSEA_MUN_C22_V6_HE_5	2221632.33	2234155.04	2245160.88	0.16	2217719.97	2217719.97	2217719.97	7.34
SHORTSEA_MUN_C23_V13_HE_1	1591809.65	1591855.05	1591923.16	0.17	1526493.15	1526493.15	1526493.15	5.64
SHORTSEA_MUN_C23_V13_HE_2	1564761.82	1564886.51	1565385.29	0.21	1536614.21	1536614.21	1536614.21	6.98
SHORTSEA_MUN_C23_V13_HE_3	1875108.52	1886805.65	1889729.94	0.18	1770199.31	1770199.31	1770199.31	5.05
SHORTSEA_MUN_C23_V13_HE_4	1675269.26	1678575.63	1683639.61	0.20	1526817.62	1526817.62	1526817.62	6.69
SHORTSEA_MUN_C23_V13_HE_5	1631311.02	1633691.82	1643215.02	0.19	1573273.06	1573273.06	1573273.06	4.60
SHORTSEA_MUN_C30_V6_HE_1	3465705.97	3467988.78	3471413.00	0.29	2950022.75	2950022.75	2950022.75	15.00
SHORTSEA_MUN_C30_V6_HE_2	3280445.83	3280923.59	3281059.50	0.31	2936017.54	2958170.88	2998787.59	15.01
SHORTSEA_MUN_C30_V6_HE_3	2982366.95	2988588.74	2998159.41	0.31	2702494.56	2702494.56	2702494.56	15.01
SHORTSEA_MUN_C30_V6_HE_4	3236303.46	3237610.70	3238221.87	0.33	3034654.37	3034654.37	3034654.37	15.00
SHORTSEA_MUN_C30_V6_HE_5	3066698.66	3081532.83	3096353.15	0.23	2945430.73	2945430.73	2945430.73	15.01
SHORTSEA_MUN_C35_V7_HE_1	3480753.15	3503066.30	3528606.13	0.53	3264245.07	3288551.97	3325397.73	15.00
SHORTSEA_MUN_C35_V7_HE_2	3467193.13	3468356.83	3472059.04	0.43	3043015.36	3043015.36	3043015.36	15.00
SHORTSEA_MUN_C35_V7_HE_3	3182368.59	3186227.73	3191364.58	0.42	3008349.15	3028384.24	3059072.67	15.01
SHORTSEA_MUN_C35_V7_HE_4	3327936.41	3328323.90	3328905.13	0.43	3094466.45	3099095.98	3117614.10	15.00
SHORTSEA_MUN_C35_V7_HE_5	3948107.09	3972851.98	4071831.53	0.49	3784395.57	3785760.94	3791222.40	15.01
SHORTSEA_MUN_C60_V13_HE_1	5878462.82	5931487.16	6013883.84	2.15	5215454.63	5238052.37	5259528.81	15.01
SHORTSEA_MUN_C60_V13_HE_2	5443824.51	5455773.94	5465467.71	1.46	5127323.74	5182831.13	5217151.58	15.01
SHORTSEA_MUN_C60_V13_HE_3	5452079.49	5488967.76	5533449.95	1.17	5019760.25	5046927.42	5072193.44	15.01
SHORTSEA_MUN_C60_V13_HE_4	5731275.00	5795704.64	5858906.54	1.52	5282856.73	5371580.08	5433885.93	15.01
SHORTSEA_MUN_C60_V13_HE_5	6225939.75	6246595.37	6284144.48	1.34	5578731.95	5649159.65	5720349.85	15.01
SHORTSEA_MUN_C80_V20_HE_1	7228373.81	7244876.99	7253636.86	4.30	6940586.81	6966300.27	6980802.69	15.02
SHORTSEA_MUN_C80_V20_HE_2	7183394.16	7232755.78	7271518.85	2.92	6768752.10	6798215.54	6847424.25	15.01
SHORTSEA_MUN_C80_V20_HE_3	6917293.24	6937672.81	6961280.12	3.18	6492245.30	6541928.66	6560944.44	15.02
SHORTSEA_MUN_C80_V20_HE_4	7907856.07	7932345.80	7962844.62	4.82	7516349.11	7604514.61	7674188.88	15.02
SHORTSEA_MUN_C80_V20_HE_5	7543369.26	7606816.98	7653860.95	2.86	7003010.40	7117798.41	7258518.33	15.02
SHORTSEA_MUN_C100_V30_HE_1	8691100.26	8716776.36	8730091.34	4.88	8370664.81	8381207.09	8398292.36	15.02
SHORTSEA_MUN_C100_V30_HE_2	8802114.22	8824225.24	8849771.38	6.20	8446069.29	8472523.45	8496617.38	15.03
SHORTSEA_MUN_C100_V30_HE_3	8355857.03	8392497.32	8409510.95	5.01	8052665.81	8068517.26	8084067.25	15.02
SHORTSEA_MUN_C100_V30_HE_4	9190265.76	9218754.67	9249184.86	5.58	8648990.22	8689451.73	8728988.09	15.03
SHORTSEA_MUN_C100_V30_HE_5	9282359.56	9321631.10	9364275.66	7.12	8527892.14	8551541.91	8588877.18	15.02
SHORTSEA_MUN_C130_V40_HE_1	11410905.75	11472383.86	11566286.06	12.31	10767542.03	10827154.34	10856536.82	15.03
SHORTSEA_MUN_C130_V40_HE_2	11176148.40	11239199.31	11321575.82	13.24	10871479.48	10911573.10	10941996.75	15.03
SHORTSEA_MUN_C130_V40_HE_3	10943790.59	11086803.90	11208710.22	11.86	10535719.46	10548431.55	10563372.19	15.03
SHORTSEA_MUN_C130_V40_HE_4	11615173.88	11657419.93	11765598.40	11.97	11028948.45	11041212.77	11054915.45	15.03
SHORTSEA_MUN_C130_V40_HE_5	13574003.55	13636707.83	13709064.75	13.05	12915656.69	13083062.16	13242595.59	15.04

Table A.10: Full results on ITSRSPSO SS_FUN instances.

	HGS-O				HGS-J			
Instance	Best	Avg	Worst	T	Best	Avg	Worst	T
SHORTSEA_FUN_C8_V3_HE_1	1013289.87	1013289.87	1013289.87	0.01	973478.67	973478.67	973478.67	0.53
SHORTSEA_FUN_C8_V3_HE_2	1000249.54	1000249.54	1000249.54	0.01	914798.06	914798.06	914798.06	0.31
SHORTSEA_FUN_C8_V3_HE_3	1300323.30	1300323.30	1300323.30	0.01	1289904.93	1289904.93	1289904.93	0.34
SHORTSEA_FUN_C8_V3_HE_4	1232379.84	1232379.84	1232379.84	0.01	1224374.01	1224374.01	1224374.01	0.58
SHORTSEA_FUN_C8_V3_HE_5	1173813.43	1173813.43	1173813.43	0.01	1125733.15	1125733.15	1125733.15	0.52
SHORTSEA_FUN_C11_V4_HE_1	846368.79	846368.79	846368.79	0.02	758817.86	758817.86	758817.86	1.10
SHORTSEA_FUN_C11_V4_HE_2	823461.48	823461.48	823461.48	0.02	723187.10	723187.10	723187.10	1.19
SHORTSEA_FUN_C11_V4_HE_3	855242.11	855242.11	855242.11	0.02	806121.58	806121.58	806121.58	1.17
SHORTSEA_FUN_C11_V4_HE_4	845851.45	845851.45	845851.45	0.02	748443.91	748443.91	748443.91	1.01
SHORTSEA_FUN_C11_V4_HE_5	976626.47	976626.47	976626.47	0.02	957044.76	957044.76	957044.76	1.52
SHORTSEA_FUN_C13_V5_HE_1	1264843.54	1264843.54	1264843.54	0.02	1241876.60	1241876.60	1241876.60	1.46
SHORTSEA_FUN_C13_V5_HE_2	1374567.90	1374567.90	1374567.90	0.03	1221025.27	1221025.27	1221025.27	1.48
SHORTSEA_FUN_C13_V5_HE_3	1492717.07	1510461.57	1529961.66	0.03	1411874.93	1411874.93	1411874.93	1.56
SHORTSEA_FUN_C13_V5_HE_4	1941350.40	1946564.64	1967421.62	0.03	1864784.67	1864784.67	1864784.67	1.52
SHORTSEA_FUN_C13_V5_HE_5	2247100.91	2254073.99	2281966.33	0.03	2097682.01	2097682.01	2097682.01	1.04
SHORTSEA_FUN_C16_V6_HE_1	2520852.41	2521742.57	2522336.01	0.04	2432112.71	2432112.71	2432112.71	1.53
SHORTSEA_FUN_C16_V6_HE_2	2402558.15	2402558.15	2402558.15	0.04	2345680.37	2345680.37	2345680.37	2.32
SHORTSEA_FUN_C16_V6_HE_3	2715354.36	2748771.74	2757126.08	0.04	2544664.54	2544664.54	2544664.54	2.58
SHORTSEA_FUN_C16_V6_HE_4	2311830.40	2311830.40	2311830.40	0.04	2172664.53	2172664.53	2172664.53	2.21
SHORTSEA_FUN_C16_V6_HE_5	2144846.58	2144846.58	2144846.58	0.04	2070448.97	2070448.97	2070448.97	2.08
SHORTSEA_FUN_C17_V13_HE_1	1484383.43	1509429.37	1526126.66	0.06	1418052.45	1418052.45	1418052.45	1.26
SHORTSEA_FUN_C17_V13_HE_2	2101437.31	2115572.82	2147307.15	0.06	2068527.68	2068527.68	2068527.68	1.51
SHORTSEA_FUN_C17_V13_HE_3	1652308.55	1653046.50	1655998.31	0.05	1548716.17	1548716.17	1548716.17	1.72
SHORTSEA_FUN_C17_V13_HE_4	1778348.98	1799932.55	1832569.03	0.05	1562165.70	1562165.70	1562165.70	2.40
SHORTSEA_FUN_C17_V13_HE_5	2041369.29	2041369.29	2041369.29	0.06	1971929.46	1971929.46	1971929.46	1.35
SHORTSEA_FUN_C20_V6_HE_1	2176655.70	2176655.70	2176655.70	0.05	1856108.50	1856108.50	1856108.50	4.06
SHORTSEA_FUN_C20_V6_HE_2	2226828.87	2234747.95	2246626.56	0.06	1973277.96	1973277.96	1973277.96	4.63
SHORTSEA_FUN_C20_V6_HE_3	2130394.19	2139900.95	2154161.10	0.05	1962909.07	1962909.07	1962909.07	5.43
SHORTSEA_FUN_C20_V6_HE_4	2045942.46	2069750.50	2089906.33	0.05	1942215.44	1942215.44	1942215.44	5.11
SHORTSEA_FUN_C20_V6_HE_5	2002612.98	2063346.18	2092530.58	0.05	1886265.64	1886265.64	1886265.64	3.51
SHORTSEA_FUN_C25_V7_HE_1	2664460.06	2686684.32	2710395.16	0.07	2527108.03	2527108.03	2527108.03	7.63
SHORTSEA_FUN_C25_V7_HE_2	2358286.89	2364818.22	2369172.44	0.07	2241254.72	2241254.72	2241254.72	7.80
SHORTSEA_FUN_C25_V7_HE_3	2847843.27	2856833.10	2875043.61	0.08	2582543.73	2582543.73	2582543.73	7.22
SHORTSEA_FUN_C25_V7_HE_4	2832202.83	2839502.66	2863351.15	0.09	2611673.70	2611673.70	2611673.70	9.79
SHORTSEA_FUN_C25_V7_HE_5	2807022.62	2821260.57	2831948.04	0.09	2469514.09	2469514.09	2469514.09	8.40
SHORTSEA_FUN_C35_V13_HE_1	2200360.92	2207006.45	2221151.82	0.16	2092392.83	2092392.83	2092392.83	14.08
SHORTSEA_FUN_C35_V13_HE_2	2279906.46	2315559.33	2355640.32	0.22	2111122.32	2111122.32	2111122.32	15.06
SHORTSEA_FUN_C35_V13_HE_3	2536809.15	2543970.57	2545760.93	0.22	2234685.10	2234685.10	2234685.10	15.00
SHORTSEA_FUN_C35_V13_HE_4	3291681.46	3295385.33	3298136.24	0.23	3077623.47	3077623.47	3077623.47	14.52
SHORTSEA_FUN_C35_V13_HE_5	2434970.93	2447699.44	2461520.01	0.21	2231281.65	2231281.65	2231281.65	14.29
SHORTSEA_FUN_C50_V20_HE_1	4959478.95	4999437.15	5032759.81	0.52	4502621.25	4502621.25	4502621.25	15.00
SHORTSEA_FUN_C50_V20_HE_2	4990335.19	5070756.86	5110144.29	0.55	4564505.62	4564745.17	4565609.84	15.00
SHORTSEA_FUN_C50_V20_HE_3	4693164.11	4721783.05	4743111.04	0.59	4351682.23	4351682.23	4351682.23	15.10
SHORTSEA_FUN_C50_V20_HE_4	6458898.80	6483047.13	6500138.52	0.59	5876803.03	5876803.03	5876803.03	15.08
SHORTSEA_FUN_C50_V20_HE_5	5149788.77	5161885.27	5170384.62	0.52	4527908.94	4527908.94	4527908.94	15.01
SHORTSEA_FUN_C70_V30_HE_1	6922089.49	6942086.49	6972014.05	1.36	6233127.33	6250942.88	6278152.90	15.01
SHORTSEA_FUN_C70_V30_HE_2	7067457.12	7093249.37	7124892.98	1.47	6464719.94	6483800.13	6522069.90	15.01
SHORTSEA_FUN_C70_V30_HE_3	6778673.86	6790545.00	6796925.13	1.61	6483064.25	6551648.95	6655441.64	15.01
SHORTSEA_FUN_C70_V30_HE_4	7940062.91	7985227.79	8005660.04	1.23	6553797.05	6581736.08	6633379.10	15.01
SHORTSEA_FUN_C70_V30_HE_5	7882155.82	7919219.25	7949740.66	1.49	6801570.31	6905173.32	7019870.11	15.01
SHORTSEA_FUN_C90_V40_HE_1	9418198.92	9447534.34	9497374.31	2.71	8496852.28	8518492.33	8528574.21	15.02
SHORTSEA_FUN_C90_V40_HE_2	9057498.26	9135539.33	9221395.46	2.63	8537822.35	8557264.25	8589997.87	15.01
SHORTSEA_FUN_C90_V40_HE_3	9041295.16	9078245.42	9099256.84	2.83	7996465.96	8013130.22	8031810.71	15.01
SHORTSEA_FUN_C90_V40_HE_4	10601100.81	10676039.78	10739927.11	2.95	9171325.93	9196158.20	9219990.44	15.01
SHORTSEA_FUN_C90_V40_HE_5	9695996.12	9757978.89	9803165.26	2.80	8379441.43	8414307.00	8497138.05	15.01
SHORTSEA_FUN_C100_V50_HE_1	9487678.62	9547498.68	9597817.06	3.97	8506548.72	8517760.95	8529036.29	15.01
SHORTSEA_FUN_C100_V50_HE_2	10031729.20	10105446.30	10150788.35	3.81	8808023.39	8812980.50	8816819.45	15.02
SHORTSEA_FUN_C100_V50_HE_3	9218124.18	9233991.02	9264852.02	3.66	8206895.30	8232574.42	8244493.73	15.01
SHORTSEA_FUN_C100_V50_HE_4	10954492.57	10988571.45	11016373.07	3.56	8846071.93	8848678.06	8852706.66	15.01
SHORTSEA_FUN_C100_V50_HE_5	10387434.04	10471780.86	10504360.80	3.66	8528096.62	8532430.54	8535903.75	15.01

Table A.11: Full results on ITSRSPSO DS_MUN instances.

	HGS-O				HGS-J			
Instance	Best	Avg	Worst	T	Best	Avg	Worst	T
DEEPSEA_MUN_C7_V3_HE_1	2832934.98	2837032.14	2843177.89	0.02	2796403.09	2796403.09	2796403.09	0.45
DEEPSEA_MUN_C7_V3_HE_2	4343563.68	4343653.89	4343714.03	0.02	3791332.21	3791332.21	3791332.21	0.38
DEEPSEA_MUN_C7_V3_HE_3	3959304.61	3971245.72	4019010.17	0.02	3907563.40	3907563.40	3907563.40	0.34
DEEPSEA_MUN_C7_V3_HE_4	5419785.44	5420140.97	5420674.27	0.02	5380211.76	5380211.76	5380211.76	0.26
DEEPSEA_MUN_C7_V3_HE_5	5433367.64	5433367.64	5433367.64	0.02	5318116.07	5318116.07	5318116.07	0.50
DEEPSEA_MUN_C10_V3_HE_1	5329474.38	5332256.50	5336429.69	0.03	4837524.52	4837524.52	4837524.52	0.64
DEEPSEA_MUN_C10_V3_HE_2	6025309.09	6033663.43	6046194.94	0.03	5857393.84	5857393.84	5857393.84	0.79
DEEPSEA_MUN_C10_V3_HE_3	6862470.07	6862470.07	6862470.07	0.03	6700231.77	6700231.77	6700231.77	0.79
DEEPSEA_MUN_C10_V3_HE_4	6879770.41	6891105.48	6893939.25	0.03	6729034.59	6729034.59	6729034.59	1.14
DEEPSEA_MUN_C10_V3_HE_5	6434613.79	6436406.76	6443578.63	0.03	5925542.44	5925542.44	5925542.44	1.16
DEEPSEA_MUN_C15_V4_HE_1	9928441.28	9933383.12	9940795.88	0.06	9556020.40	9556020.40	9556020.40	4.02
DEEPSEA_MUN_C15_V4_HE_2	8499671.69	8505419.18	8522950.50	0.08	7261934.50	7261934.50	7261934.50	3.92
DEEPSEA_MUN_C15_V4_HE_3	9459347.46	9459347.46	9459347.46	0.06	8935752.07	8935752.07	8935752.07	4.01
DEEPSEA_MUN_C15_V4_HE_4	8116517.31	8119594.79	8124211.01	0.08	7815187.29	7815187.29	7815187.29	3.87
DEEPSEA_MUN_C15_V4_HE_5	6986863.91	6990466.94	6995871.48	0.07	6451325.93	6451325.93	6451325.93	3.49
DEEPSEA_MUN_C18_V5_HE_1	20861231.57	20865349.92	20881823.33	0.10	20397321.43	20397321.43	20397321.43	6.42
DEEPSEA_MUN_C18_V5_HE_2	8622981.65	8624615.10	8625023.46	0.13	8524460.98	8524460.98	8524460.98	6.94
DEEPSEA_MUN_C18_V5_HE_3	10534483.76	10534483.76	10534483.76	0.10	9598514.21	9598514.21	9598514.21	5.29
DEEPSEA_MUN_C18_V5_HE_4	14338840.21	14341301.93	14344994.50	0.10	12172843.29	12172843.29	12172843.29	5.26
DEEPSEA_MUN_C18_V5_HE_5	22799028.03	22805607.35	22809993.56	0.10	19109452.90	19109452.90	19109452.90	5.59
DEEPSEA_MUN_C22_V6_HE_1	36556153.73	36567936.81	36615069.12	0.15	30899720.75	30899720.75	30899720.75	8.22
DEEPSEA_MUN_C22_V6_HE_2	30286683.75	30291516.69	30300260.04	0.14	26662969.12	26662969.12	26662969.12	7.66
DEEPSEA_MUN_C22_V6_HE_3	32898441.97	33054543.27	33111467.28	0.16	24188291.48	24188291.48	24188291.48	7.83
DEEPSEA_MUN_C22_V6_HE_4	27576546.47	27658465.68	27794203.77	0.16	24883538.36	24883538.36	24883538.36	12.32
DEEPSEA_MUN_C22_V6_HE_5	35778633.14	35778633.14	35778633.14	0.17	33173180.68	33173180.68	33173180.68	9.39
DEEPSEA_MUN_C23_V13_HE_1	31816765.90	32018195.88	32144681.38	0.18	28622900.41	28622900.41	28622900.41	3.48
DEEPSEA_MUN_C23_V13_HE_2	19402202.24	19427239.32	19441961.05	0.21	17646501.30	17646501.30	17646501.30	7.40
DEEPSEA_MUN_C23_V13_HE_3	17807535.83	17809510.87	17810827.57	0.16	17081323.45	17081323.45	17081323.45	4.76
DEEPSEA_MUN_C23_V13_HE_4	24119863.21	24415740.53	24612992.08	0.22	19896624.07	19896624.07	19896624.07	5.25
DEEPSEA_MUN_C23_V13_HE_5	26285987.03	26285987.03	26285987.03	0.16	23880887.01	23880887.01	23880887.01	5.98
DEEPSEA_MUN_C30_V6_HE_1	13192750.87	13411379.10	13735707.64	0.22	11545020.82	11545020.82	11545020.82	15.04
DEEPSEA_MUN_C30_V6_HE_2	11853569.65	11880631.49	11916539.42	0.30	10562790.91	10583661.56	10667144.15	15.12
DEEPSEA_MUN_C30_V6_HE_3	15520766.33	15520766.33	15520766.33	0.28	14401537.42	14401537.42	14401537.42	15.00
DEEPSEA_MUN_C30_V6_HE_4	15122149.17	15443682.33	15664433.86	0.26	12523728.83	12523728.83	12523728.83	15.00
DEEPSEA_MUN_C30_V6_HE_5	17865635.85	17871757.09	17888381.21	0.29	17130423.34	17158645.50	17200978.74	15.13
DEEPSEA_MUN_C35_V7_HE_1	41120911.34	41453693.45	41936142.87	0.40	39110149.15	39187684.61	39497826.46	15.01
DEEPSEA_MUN_C35_V7_HE_2	36730160.86	36770874.11	36798016.27	0.38	32997272.90	33674825.87	33967631.63	15.00
DEEPSEA_MUN_C35_V7_HE_3	37198284.64	37474462.10	37560042.28	0.39	36024002.98	36142317.79	36615577.01	15.01
DEEPSEA_MUN_C35_V7_HE_4	41887364.89	42072301.94	42208105.88	0.39	36177521.34	36313285.78	36612779.43	15.01
DEEPSEA_MUN_C35_V7_HE_5	42408663.93	42451876.38	42570002.50	0.29	38869404.09	39057832.16	39811544.43	15.00
DEEPSEA_MUN_C60_V13_HE_1	57788220.34	58059969.50	58380187.51	2.20	55344983.98	55996020.49	56639990.39	15.03
DEEPSEA_MUN_C60_V13_HE_2	52606575.13	52705610.26	52812781.57	1.28	49163913.78	49947632.96	50567023.14	15.03
DEEPSEA_MUN_C60_V13_HE_3	66271207.88	66682884.54	67257653.16	1.40	57034176.35	58199470.04	59761093.98	15.02
DEEPSEA_MUN_C60_V13_HE_4	66808214.00	66986290.33	67159088.18	1.60	60354499.70	62698383.07	64274583.20	15.02
DEEPSEA_MUN_C60_V13_HE_5	68076353.64	68816238.94	71604688.80	1.63	65084662.89	65905176.95	67368482.75	15.03
DEEPSEA_MUN_C80_V20_HE_1	88731911.69	89499990.91	90097013.44	3.76	80853704.69	81918794.69	83193236.49	15.02
DEEPSEA_MUN_C80_V20_HE_2	89415537.11	91843621.09	94471756.56	5.85	74673119.13	75673065.80	76664095.35	15.02
DEEPSEA_MUN_C80_V20_HE_3	95689409.99	98072618.73	99353638.44	3.33	84216362.73	85548557.28	86445952.57	15.01
DEEPSEA_MUN_C80_V20_HE_4	91546061.61	94373568.12	97370988.15	2.08	82647679.72	83723994.02	84633747.02	15.01
DEEPSEA_MUN_C80_V20_HE_5	100242467.77	100651999.05	100857270.25	2.67	78081248.74	79502070.84	80795403.87	15.01
DEEPSEA_MUN_C100_V30_HE_1	100967995.27	103497858.13	105170530.57	8.24	87494720.89	88677707.26	89500878.47	15.03
DEEPSEA_MUN_C100_V30_HE_2	107732715.23	107966781.37	108454325.09	8.57	92825444.90	93326242.30	94079234.97	15.03
DEEPSEA_MUN_C100_V30_HE_3	101893212.69	102036471.57	102169144.54	8.29	89011667.61	90576660.31	92795125.32	15.04
DEEPSEA_MUN_C100_V30_HE_4	103325308.70	104373270.20	105588295.97	7.97	95812819.93	96578411.48	97982469.21	15.02
DEEPSEA_MUN_C100_V30_HE_5	103828287.57	104070943.93	104563902.77	8.74	95707451.07	96429315.55	97842987.40	15.04
DEEPSEA_MUN_C130_V40_HE_1	81394029.76	82263721.18	83176959.82	15.08	85810122.92	89150385.88	92330083.81	15.10
DEEPSEA_MUN_C130_V40_HE_2	76417396.46	77244364.56	78908852.60	15.00	79391700.04	80317914.50	81728734.24	15.03
DEEPSEA_MUN_C130_V40_HE_3	82243158.53	83086879.83	83811672.90	15.00	89057313.32	91532007.87	95685020.58	15.05
DEEPSEA_MUN_C130_V40_HE_4	61769933.71	62234530.43	62859635.41	15.01	69758667.83	73609751.57	76665599.06	15.06
DEEPSEA_MUN_C130_V40_HE_5	80097964.48	80840382.64	83149927.99	15.30	90786248.30	94424777.94	98578728.72	15.06

Table A.12: Full results on ITSRSPSO DS_FUN instances.

	HGS-O				HGS-J			
Instance	Best	Avg	Worst	T	Best	Avg	Worst	T
DEEPSEA_FUN_C8_V3_HE_1	7053629.77	7053629.77	7053629.77	0.01	5176264.79	5176264.79	5176264.79	0.55
DEEPSEA_FUN_C8_V3_HE_2	7633883.10	7633883.10	7633883.10	0.01	7411131.41	7411131.41	7411131.41	0.40
DEEPSEA_FUN_C8_V3_HE_3	2603121.04	2603121.04	2603121.04	0.01	2603121.04	2603121.04	2603121.04	0.27
DEEPSEA_FUN_C8_V3_HE_4	5608794.59	5608794.59	5608794.59	0.01	5468590.60	5468590.60	5468590.60	0.72
DEEPSEA_FUN_C8_V3_HE_5	4715457.02	4715457.02	4715457.02	0.01	4633069.90	4633069.90	4633069.90	0.85
DEEPSEA_FUN_C11_V4_HE_1	20239020.95	20239020.95	20239020.95	0.02	20239020.95	20239020.95	20239020.95	0.74
DEEPSEA_FUN_C11_V4_HE_2	12208695.13	12208695.13	12208695.13	0.02	11874556.52	11874556.52	11874556.52	1.50
DEEPSEA_FUN_C11_V4_HE_3	13026718.56	13026718.56	13026718.56	0.02	13026718.56	13026718.56	13026718.56	1.19
DEEPSEA_FUN_C11_V4_HE_4	16582971.92	16582971.92	16582971.92	0.02	11270464.22	11270464.22	11270464.22	0.93
DEEPSEA_FUN_C11_V4_HE_5	14498736.72	14498736.72	14498736.72	0.02	11408712.87	11408712.87	11408712.87	1.45
DEEPSEA_FUN_C13_V5_HE_1	8039849.57	8039849.57	8039849.57	0.03	7953555.54	7953555.54	7953555.54	1.93
DEEPSEA_FUN_C13_V5_HE_2	8435738.12	8435738.12	8435738.12	0.03	8232260.88	8232260.88	8232260.88	1.44
DEEPSEA_FUN_C13_V5_HE_3	7136618.66	7136618.66	7136618.66	0.03	6856436.04	6856436.04	6856436.04	1.87
DEEPSEA_FUN_C13_V5_HE_4	9741715.84	9741715.84	9741715.84	0.03	9286602.75	9286602.75	9286602.75	1.33
DEEPSEA_FUN_C13_V5_HE_5	6793423.45	6825675.34	6874053.17	0.02	6689508.72	6689508.72	6689508.72	1.55
DEEPSEA_FUN_C16_V6_HE_1	29871927.48	29871927.48	29871927.48	0.04	24627424.31	24627424.31	24627424.31	2.61
DEEPSEA_FUN_C16_V6_HE_2	18846052.77	18846052.77	18846052.77	0.04	18374891.10	18374891.10	18374891.10	3.12
DEEPSEA_FUN_C16_V6_HE_3	25639454.55	25651324.60	25659237.97	0.04	19778140.50	19778140.50	19778140.50	2.26
DEEPSEA_FUN_C16_V6_HE_4	12020651.76	12115488.93	12200672.06	0.04	11601030.22	11601030.22	11601030.22	2.33
DEEPSEA_FUN_C16_V6_HE_5	18186300.47	18186300.47	18186300.47	0.04	17184445.98	17184445.98	17184445.98	2.59
DEEPSEA_FUN_C17_V13_HE_1	14178294.59	14178294.59	14178294.59	0.06	13222105.46	13222105.46	13222105.46	1.77
DEEPSEA_FUN_C17_V13_HE_2	7072317.27	7072546.78	7073464.81	0.05	6115054.90	6115054.90	6115054.90	2.03
DEEPSEA_FUN_C17_V13_HE_3	8511082.46	8535559.27	8551877.14	0.05	7371359.05	7371359.05	7371359.05	2.00
DEEPSEA_FUN_C17_V13_HE_4	7565230.47	7573754.57	7586540.71	0.06	6978841.51	6978841.51	6978841.51	1.79
DEEPSEA_FUN_C17_V13_HE_5	8228314.78	8335527.39	8365885.37	0.06	7675819.79	7675819.79	7675819.79	2.76
DEEPSEA_FUN_C20_V6_HE_1	12334463.96	12338383.33	12344262.39	0.05	10726291.11	10726291.11	10726291.11	4.09
DEEPSEA_FUN_C20_V6_HE_2	11465951.24	11465951.24	11465951.24	0.06	10532603.30	10532603.30	10532603.30	4.97
DEEPSEA_FUN_C20_V6_HE_3	13194245.08	13246779.68	13259913.33	0.05	12539855.50	12539855.50	12539855.50	5.09
DEEPSEA_FUN_C20_V6_HE_4	11245339.50	11433961.14	11488426.10	0.05	10582247.98	10582247.98	10582247.98	6.57
DEEPSEA_FUN_C20_V6_HE_5	13155920.32	13155920.32	13155920.32	0.06	11495942.21	11495942.21	11495942.21	4.42
DEEPSEA_FUN_C25_V7_HE_1	16443589.66	16600861.22	16734299.10	0.08	14551679.99	14551679.99	14551679.99	8.51
DEEPSEA_FUN_C25_V7_HE_2	14444533.33	14444533.33	14444533.33	0.08	13818829.77	13818829.77	13818829.77	10.15
DEEPSEA_FUN_C25_V7_HE_3	12715248.95	13019002.95	13141212.27	0.08	11462608.41	11462608.41	11462608.41	7.33
DEEPSEA_FUN_C25_V7_HE_4	15055615.59	15100939.54	15193891.95	0.08	13531763.52	13531763.52	13531763.52	11.25
DEEPSEA_FUN_C25_V7_HE_5	17578324.75	17632385.27	17668425.61	0.08	16916916.83	16916916.83	16916916.83	12.05
DEEPSEA_FUN_C35_V13_HE_1	34091943.46	34187730.92	34361554.67	0.24	32190876.48	32190876.48	32190876.48	15.00
DEEPSEA_FUN_C35_V13_HE_2	34979535.48	34985948.91	34987552.27	0.29	32340250.02	32340250.02	32340250.02	15.00
DEEPSEA_FUN_C35_V13_HE_3	36367935.91	36407859.64	36473267.83	0.26	31250194.55	31250194.55	31250194.55	15.00
DEEPSEA_FUN_C35_V13_HE_4	38326874.77	38358190.68	38391696.40	0.26	34362295.26	34362295.26	34362295.26	15.21
DEEPSEA_FUN_C35_V13_HE_5	38982034.94	38997536.90	39007871.54	0.24	36601759.93	36601759.93	36601759.93	15.05
DEEPSEA_FUN_C50_V20_HE_1	31714106.40	31722174.05	31744213.40	0.65	26868909.98	26868909.98	26868909.98	15.19
DEEPSEA_FUN_C50_V20_HE_2	29780153.21	29847181.62	29891070.44	0.61	24148564.29	24189961.67	24284477.25	15.00
DEEPSEA_FUN_C50_V20_HE_3	30485469.66	30487184.25	30489756.13	0.61	25726501.72	25726501.72	25726501.72	15.00
DEEPSEA_FUN_C50_V20_HE_4	32341061.45	32479423.06	32532799.55	0.65	27960578.46	27961127.15	27963321.90	15.01
DEEPSEA_FUN_C50_V20_HE_5	31327248.97	31596551.41	31845446.44	0.62	27224627.19	27224627.19	27224627.19	15.01
DEEPSEA_FUN_C70_V30_HE_1	96776778.48	96994180.55	97318603.01	1.62	84940529.78	85185022.67	85510551.59	15.01
DEEPSEA_FUN_C70_V30_HE_2	101233577.26	101390265.72	101513036.32	1.55	78563704.63	79386397.42	80093285.14	15.01
DEEPSEA_FUN_C70_V30_HE_3	119659725.92	120426620.47	120986171.23	1.46	104515448.02	104975140.31	105241833.49	15.01
DEEPSEA_FUN_C70_V30_HE_4	120159116.50	120200867.22	120265334.01	1.44	98372657.52	98802378.04	99322610.04	15.01
DEEPSEA_FUN_C70_V30_HE_5	116358602.13	116495270.59	116555689.96	1.32	92954769.67	93108353.19	93218243.17	15.01
DEEPSEA_FUN_C90_V40_HE_1	135458583.60	136344656.25	136926558.29	3.55	113501619.49	114461536.19	115128848.64	15.03
DEEPSEA_FUN_C90_V40_HE_2	139376523.52	139751577.58	140496167.11	3.17	122916831.37	123669276.57	124164156.64	15.01
DEEPSEA_FUN_C90_V40_HE_3	158592379.92	158837939.94	158932796.87	3.41	139473007.91	140391743.60	141452184.68	15.01
DEEPSEA_FUN_C90_V40_HE_4	163114981.16	163654646.82	163974917.73	2.77	133915724.73	134045556.85	134128397.72	15.01
DEEPSEA_FUN_C90_V40_HE_5	146077540.41	146563253.47	146768867.44	3.27	124320546.16	125537789.39	126606396.12	15.02
DEEPSEA_FUN_C100_V50_HE_1	149978246.96	150115446.68	150271245.82	4.54	123098500.67	124657435.94	125566932.64	15.03
DEEPSEA_FUN_C100_V50_HE_2	149807415.88	150236003.32	151136510.96	4.78	129416636.97	130157542.63	130484081.45	15.02
DEEPSEA_FUN_C100_V50_HE_3	161528601.40	161717042.37	161835645.19	4.42	133268424.23	133856208.44	134367773.97	15.03
DEEPSEA_FUN_C100_V50_HE_4	166670427.42	167209515.85	167754655.36	4.19	133445916.93	133723245.09	134016416.15	15.02
DEEPSEA_FUN_C100_V50_HE_5	166410544.91	166646811.27	166735760.17	4.22	144175327.73	144268272.80	144451106.78	15.02