Optimising commercial port operations through high-level information fusion

Ashwin Panchapakesan*

School of Electrical Engineering and Computer Science,
University of Ottawa,
Ottawa, Canada
Email: apanc006@uottawa.ca
*Corresponding author

Rami Abielmona

Research and Engineering,
Larus Technologies,
Ottawa, Canada
Email: rami.abielmona@larus.com

Emil Petriu

School of Electrical Engineering and Computer Science,
University of Ottawa,
Ottawa, Canada
Email: petriu@uottawa.ca

Abstract: In order to remain profitable, commercial maritime ports must maintain high throughput of inbound vessels. The gantry cranes that load and unload the vessels are the primary point of interaction between a vessel and the port, which cause a critical bottleneck in the process flow. Errors in this segment of the process cause cascading delays which ultimately cause vessel service backlogs, extending to logistical delays in moving shipping containers across land and rail as well. This is to the detriment of the ports, which lose popularity among shipping lines and may even be fined for causing sub-optimal delays. This work expands on prior work in using a multi-objective genetic algorithm to optimise the parameters of a fuzzy system which controls port-side resource deployment. In contrast to existing solutions, this resource deployer is able to function online and adapt to real-world faults while still maintaining superior performance as compared to industry practice. Further, proposals to expand or reduce the port-side infrastructure are computed.

Keywords: information fusion; multi-objective optimisation; genetic algorithms; fuzzy systems; data mining.

Reference to this paper should be made as follows: Panchapakesan, A., Abielmona, R. and Petriu, E. (2021) 'Optimising commercial port operations through high-level information fusion', *Int. J. Logistics Systems and Management*, Vol. 40, No. 2, pp.242–265.

Copyright © 2021 Inderscience Enterprises Ltd.

Biographical notes: Ashwin Panchapakesan is a graduate student at the Department of Computer Science, School of Electrical Engineering and Computer Science University of Ottawa, Ottawa, Canada.

Rami Abielmona is the VP of Research and Engineering, Larus Technologies, Ottawa, Canada.

Emil Petriu is a tenured faculty member at the Department of Electrical Engineering, School of Electrical Engineering and Computer Science University of Ottawa, Ottawa, Canada.

This paper is a revised and expanded version of a paper entitled 'Optimizing maritime vessel service time with adaptive quay crane deployment through level 4 hard-soft information fusion' presented at FUSION '19, Ottawa, Canada, July 2019.

1 Introduction

Maritime trade comprises nearly nine tenths of global trade (Cheraghchi et al., 2017). As a result, commercial maritime ports form a choke point in the overarching global trade process. Port operations are varied and include controlling the surrounding water space to coordinate marine traffic, internal security, repair and maintenance, etc. The primary point of contact of a container ship with the port, from the perspective of global trade is the ship-to-shore gantry crane (or quay crane) which unloads containers from the inbound vessel and loads new containers onto the outbound vessel. This process of unloading and loading containers from a vessel is referred to here as a vessel service. Consequently, the time required to complete vessel service is the vessel service time. This vessel service is performed by quay cranes, which require operating personnel as well as other supporting equipment and personnel such as internal shunt trucks, fork lifts, spotters, checkers, etc. all of which make up the deployable port side resources.

A naive method to ensure maximised service throughput is to maximise resource deployment so that all incoming vessel service loads are handled at the port's maximum bandwidth. However, deploying resources comes at the cost of equipment operating costs (including machine wear), and personnel wages. More importantly (from an optimisation perspective), while machine operating costs scale very well with incoming loads (since there are no costs associated with non-operating equipment), personnel are deployed on a per-shift basis which requires the system to bear the cost of an entire shift's crew wage even if the equipment is used only for a fraction of that shift.

A static vessel arrival schedule allows for the analytic optimisation of the port-side resource deployment to handle the incoming vessel service loads. However, since schedules are temporally continuous, not bound by any end time, and subject to unexpected change, such offline, analytical optimisation would be rigid, brittle, and expensive in its need to be performed on each schedule change. A more dynamic approach with an optimised fuzzy system would be less computationally expensive to deploy and more flexible in its ability to adapt to changes in vessel arrival and therefore service schedules.

Such continually adaptive systems fall under level 4 of the data fusion model proposed by the Joint Directors of Laboratories' (JDL's) Data Fusion Information Group (DFIG) (Blasch et al., 2012) and is used in this work. The resource deployment is controlled by a fuzzy system while the parameters of this fuzzy system are optimised by a genetic algorithm (GA). During the process of this optimisation, the fuzzy system is evaluated on the total time required to service all vessels and on the number of port-side personnel crews required to perform the vessel service given the deployment.

The remainder of this paper is organized as follows: some introductory background material and prior work are presented in Section 2 and the methodology used is specified in Section 3. The results of the proposed methodology are presented in Section 4 and some concluding remarks and directions of future work are presented in Section 5.

2 Previous work

Presented in this section are brief introductions to the relevant background material and observations of prior solutions to similar problems.

2.1 Data fusion

Methods to improve optimisation processes fall under level 4 (process refinement) of the data fusion model of the DFIG of the JDLs. This level of the data fusion model is an abstraction over level 3 (impact assessment and course of action generation) and works to improve the processes at these levels with regards to their efficacy of their outputs and well as the efficiency with which they are generated. This optimization is guided by measures of effectiveness (MOEs) and measures of performance (MOPs), which describe the optimality of a given process-optimised solution (Blasch et al., 2012).

2.2 Fuzzy systems

Fuzzy systems allow for the implementation of control systems using precise digital measurements which may fall into imprecisely, linguistically described rules. For instance, the rule "`if it is cold, increase the temperature on the thermostat`" is a clear linguistic instruction with no precise machine-operable counterpart. As a result, layers of fuzzification and defuzzification are built to respectively map precise values measured by sensors to linguistically described sets such as '`cold`' and '`hot`' and map imprecise instructions such as '`increase the temperature on the thermostat`' to a precise, machine actionable value such as '+0.3°C'.

As such, a fuzzy system is comprised of three components: a fuzzifier, a defuzzifier, and an inference engine. The Mamdani fuzzy system used in this work is described in this section.

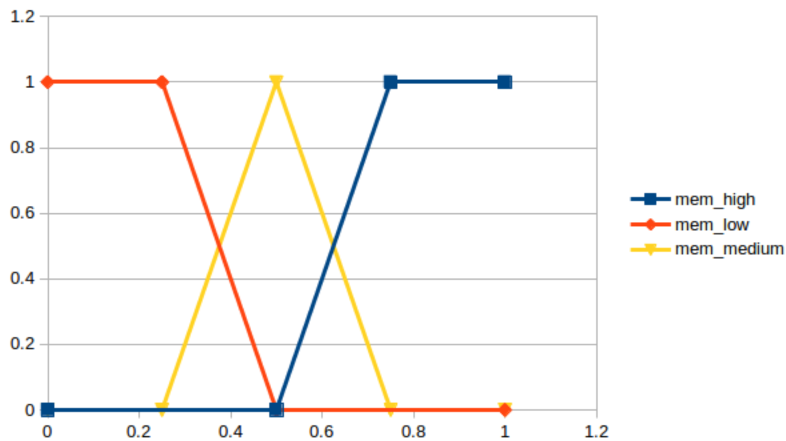
The fuzzy system is optimised to determine resource deployment at a given instance. Therefore, the inputs to the fuzzy system are

1 the current container processing load

2 the difference in the container processing load between the current and the previous shift.

These inputs determine whether the load is light or heavy, and whether it is increasing or decreasing. Since the personnel deployment can be determined only on a per-shift basis, the resource deployment provided by the fuzzy system must be recalculated at a frequency no greater than at each shift change. An example of the degree of membership of the measured load is computed to each of the linguistically descriptive sets of 'low', 'medium', and 'high' load as shown in Figure 1, and the parameters of this membership function are optimised by the GA described in Subsection 2.3. Additionally, the change in load is computed at the end of each shift as the difference between the number of containers to be processed at the start of the shift, and the sum of the number of containers remaining to be processed and the shift and the number of containers to be processed on any incoming vessels during this shift. The memberships of the change in processing load are also mapped to the linguistic sets 'negative change', 'minimal change', and 'positive change'.

Figure 1 Fuzzy membership functions to 'low', 'medium', and 'high' load (see online version for colours)



Once these memberships are computed, the fuzzy inference engine computes the amplitude of resource deployment based on the following rule set:

1. If `delay` is low and `delta_delay` is low, decrease deployment.
2. If `delay` is medium and `delta_delay` is low, decrease deployment.
3. If `delay` is high and `delta_delay` is low, maintain deployment.
4. If `delay` is low and `delta_delay` is unchanging, decrease deployment.
5. If `delay` is medium and `delta_delay` is unchanging, maintain deployment.
6. If `delay` is high and `delta_delay` is unchanging, increase deployment.
7. If `delay` is low and `delta_delay` is high, increase deployment.
8. If `delay` is medium and `delta_delay` is high, increase deployment.
9. If `delay` is high and `delta_delay` is high, increase deployment.

Having calculated the fuzzy membership functions and the inference engine, the fuzzy system uses a centre-of-gravity calculation to compute the resource deployment.

2.3 Multi-objective GAs

GAs haven been used in numerous optimisation applications (Xu et al., 2019; Teske et al., 2017; Suri and Vijay, 2019; Rey Horn et al., 1994; Li et al., 2019; Davis, 1991). These require a descriptive structure of the solution, and a method for creating multiple such solutions to form a population of randomly generated individuals; and a method of evaluating a given encoded solution. Once evaluated, the individuals are ranked on a [0, 1] scale and are selected in some way that correlates with their fitness relative to the other individuals in the population. Once individuals have been selected, parts of their encoded solution structures are copied over to form new 'child' individuals that become part of the next generation of the population. This process is repeated for many generations until some termination condition is met. Typically, a termination condition describes a minimum solution quality and/or a time limit.

It is important to note that a multi-objecive GA (MOGA) differs from a regular GA in that the range of its fitness function lies in a multi-dimensional space. Therefore, the notion of non-dominance is important to discuss. Two solutions are non-domminated if one individual's fitness values do not outperform the other's in all dimensions. For example, suppose individuals A, and B are evaluated in a two-dimensional fitness plane, with fitness measures $f(A) =< 1, 2 >$ and $f(B) =< 2, 1 >$ where each fitness objective is required to be minimised. Then, neither A nor B dominates the other as they each outperform the other in one of the two objectives. On the other hand, if individuals C and D had fitness measures $f(C) =< 2, 2 >$ and $f(D) =< 2, 1 >$, then it is clear that D dominates C since D's fitness values are at least as optimal as C's on each fitness dimension. Therefore, selection and fitness ranking in a MOGA are performed on a multi-dimensional fitness landscape with non-dominated fitness 'fronts'. GAs and MOGAs have also been used to optimise fuzzy systems (Teske et al., 2017), to optimise the various parameters of the fuzzy system.

Methodologies used in such optimisations are adopted in this work and the specifics of the GA used in this study are described in Subsection 3.2.

2.4 Resource deployment

Fuzzy controllers have been used to dynamically allocate resources in large-scale job scheduling environments with strict job completion deadlines Cheng et al. (2015). This was performed by phrasing the job scheduling problem as a prediction optimisation problem of job completion times given resource availability. The estimator that predicts job completion time is formulated as a fuzzy system that considers the job requirements and resource availability as inputs to output a deadline-aware scheduling for a given job. Others have suggested a continuously evolving fuzzy system to circumvent the limitations of rule bases fixed at design time, that therefore fall short of changing objectives Jamshidi et al. (2016). While this methodology is impressive and powerful, the authors note that it is in excess of the requirements of problems similar to the one studied here (deploying existing resources to meet queued demand); and approaches from classical0 control theory will still suffice. Wang et al. (2011) do indeed use such a fuzzy system to adaptively deploy resources to handle database load within a virtual environment to handle incoming query loads. This system also updates *online*, making it reactively adaptive to dynamic workloads.

Existing solutions to resource deployment in the maritime space suffer from being *offline*, i.e., they require *a priori* knowledge of incoming vessel schedules and their respective service load requirements. Some of these have been formulated as mixed integer programming (MIP) problems (Zhang et al., 2002), while others have attempted a job scheduling formulation with a homogenouos pool of processors (Peterkofsky and Daganzo, 1990). Both methods require prior knowledge of incoming vessel schedules and service requirements, which is inconsistent with real-world practice. Moreover, these methods are unable to adapt to changing vessel schedules and operational faults such as delays in vessel arrival times [often caused by unfavorable weather conditions, etc. (Cheraghchi et al., 2017)]. Finally, GAs have been used in optimising resource deployment and scheduling Wesolkowski et al. (2014), but only in an offline fashion yet again. The methodology presented in this work proves to be robust against vessel schedule changes, while maintaining high service throughput and simultaneously minimising operational cost.

3 Methodology

The methodologies employed in this study to acquire data and perform the optimisation are discussed in this section.

3.1 Dataset

Maritime vessels are required by law to broadcast information about their geospatial location, the type of cargo onboard, speed, destination, vessel draught, etc. including a unique identifier, in a standardised automated information system (AIS) data packet (Perez et al., 2009). These are then gathered by terrestrial or satellite receivers. Correlating each vessel's AIS packets yields a comprehensive track of the vessel's voyage over time. Plotting this on a map (using any geographic information system) shows the time at which a vessel arrived and departed a port. Filtering for such segments in vessel tracks shows the ground truth vessel service time, mined from real-world data. Further, filtering these tracks by proximity to known ports such as the Port of Halifax, and the Port of Hong Kong gives the vessel service time specific to each port, which can then be used by the optimisation algorithm described in Subsection 3.2. Still, this does not account for vessels that may simply pass by the port, without being serviced, which is why an additional filter is used to capture only vessels that are stationery in the given time period, as having a vessel speed of 0.5 knots or lower (Abualhaol et al., 2018).

Finally, the vessel's draught before and after service are computed and the vessel's size (length and width) is read from the AIS packets. With this information, the total volume of displaced water is computed, which when multiplied by the density of water, gives the mass of displaced sea water. In the absence of any further information describing the mass of fuel and ballast on board the vessel, the mass of the displaced water is assumed to be the mass of the containers and cargo on board. When divided by the mass of a cargo-filled shipping container drawn from the normal distribution

$\mathcal{N}(\mu = 75, \sigma = 10)$, this yields an estimate of the number of containers on board the vessel. Adding together the number of containers that entered the port and the number of containers that left (making the assumption that all of a vessel's containers are unloaded and all the containers it leaves port with are new containers that were loaded at port) gives the number of containers that comprised vessel service.

Given that a container is serviced by a single crane in two minutes (Zhang et al., 2002), it is possible to sum the serviced containers per shift to mine the real-world resource deployment, to use as a benchmark in optimisation.

It is also known that ports internally publish daily situation reports describing the day's loads and projected loads for the following day. Using this information source could prove to be useful in predicting upcoming service loads and lead to more accurate resource deployment. However, this comes with the computational cost of requiring the addition of a natural language processing (NLP) module to extract information from soft, unstructured, daily logs. The procedure to perform this information extraction is described in Subsection 3.2.

Finally, AIS data is gathered from multiple sources and is subsequently fused and correlated, as previously discussed. This affords the opportunity to determine the usefulness of each data source independently of the others. In order to evaluate the usefulness of a data source, data from different sources are ingested separately for an independent run of the optimisation. The results of these independent runs of the optimisation are then compared against the results from the optimisation performed with the combined data from all sources. The expectation is that the performance of the optimisation using individual data sources would perform sub optimally as compared to using the combined data from all sources.

3.2 Optimisation

The GA used to optimise the fuzzy membership functions described in Figure 1 is specified in this section. The GA uses individuals with three chromosomes each, to specify the membership functions to low, medium, and high vessel service loads. Further, each individual that encodes a fully specified fuzzy system is evaluated by simulating known incoming vessel loads to determine the optimality with which the vessel service is performed.

3.2.1 Individual structure

Each individual is encoded with three chromosomes. Each chromosome respectively encodes for membership to low, medium, and high container load. An example of this is shown in Table 1. Each of these individuals specifies the trapezoidal membership function of a given container load to 'low', 'medium', or 'high' load. The actualisation of these chromosomal values to the membership functions are governed by equations (1), (2) and (3).

$$\text{mem\_low}(x) = \begin{cases} 1 & x < A \\ \frac{100-100x}{17} & A \leq x < B \\ 0 & B \leq x \end{cases} \tag{1}$$

$$\text{mem\_med}(x) = \begin{cases} 0 & x < A \\ \frac{6}{|x-\frac{1}{2}|} & A \leq x < B \\ 0 & C \leq x \end{cases} \tag{2}$$

$$\text{mem\_high}(x) = \begin{cases} 1 & x < B \\ \frac{100x-100}{17} & B \leq x < C \\ 0 & C \leq x \end{cases} \tag{3}$$

Table 1 Chromosomal structure for trapezoidal individual

	A	*B*	*C*	*D*
$\mu_{\texttt{low}}$	0.33	0.5	0.66	1
$\mu_{\texttt{medium}}$	0.33	0.5	0.66	1
$\mu_{\texttt{high}}$	0.33	0.5	0.66	1

3.2.2 Fitness

The fitness of an individual in the MOGA is a point on a two-dimensional fitness landscape describing the number of equipment and personnel crews to be used and the total time required to service all vessels. Both of these objectives must be minimised. Given an individual that describes the fuzzy membership function, the fuzzy system is created, and the mined vessel arrival is simulated. Additionally, daily situation reports are generated as part of the simulation, by sampling the language used in known situation reports. Therefore, if the load is projected to increase (as seen in the simulation), a situation report with exaggerative adjectives describing increased load is generated and vice versa.

3.2.3 Selection

The selection mechanism yields two individuals for mating operations (crossover and mutation). Since the optimal fitness is unknown, the individuals in the population are sorted into fitness fronts. Each individual is selected independently at random from a given fitness front. The fitness front itself is selected in such a way as to correlate with the front's relative optimality. Therefore, the probability of a front being selected (from which an individual will be selected) is $\frac{i}{i+1}$ for front i. This gives each front a linearly scaled probability of being selected, with front 1 (the Pareto front) having probability 0.5 of being selected.

3.2.4 Crossover

The two individuals returned by the selection mechanism are then mated. Since the fitness describes the optimality of the individual, the child individual would benefit from an interpolated encoding that is influenced by both parent individuals, weighted by their respective fitness measures. However, since weighting is a scalar operation and the fitness measure resides in a two dimensional space, a scalarisation technique is used to plot the fitness in a one dimensional space [as seen in equations (4) and (5)]. The resultant value of the child individual is therefore the fitness weighted average of the corresponding values of the parent individuals. An example of this is seen in Table 2. Note that since this is an averaging operation, the result of crossover is one child individual, as opposed to classical crossover operators which yield two or more child individuals.

$$\overline{F_i} = \frac{\sum_{x=1}^{|P|} F_i P_x}{|P|} \tag{4}$$

$$F_r(P_x) = \frac{1}{2} \cdot \left[\sum_{i=1}^{2} F_i(P_x) - \overline{F_i} \right] \tag{5}$$

where

F_r is the relative fitness measure

$F_i(P_x)$ is the fitness of individual P_x along objective i

P is the population of all individuals

Table 2 Example crossover

	(a) Parent individual P_1 (F_r = 1)			
	A	*B*	*C*	*D*
$\mu_{\texttt{low}}$	0.33	0.5	0.66	1
$\mu_{\texttt{medium}}$	0.33	0.5	0.66	1
$\mu_{\texttt{high}}$	0.33	0.5	0.66	1
	(b) Parent individual P_2 (F_r = 3)			
	A	*B*	*C*	*D*
$\mu_{\texttt{low}}$	0	0.5	0.66	1
$\mu_{\texttt{medium}}$	0	0.5	0.66	1
$\mu_{\texttt{high}}$	0	0.5	0.66	1
	(c) Child individual			
	A	*B*	*C*	*D*
$\mu_{\texttt{low}}$	0.0825	0.5	0.66	1
$\mu_{\texttt{medium}}$	0.0825	0.5	0.66	1
$\mu_{\texttt{high}}$	0.0825	0.5	0.66	1

3.2.5 Mutation

The mutation operator randomly changes the value of a randomly selected element in the individual's chromosomal structure. A random chromosome is first chosen, and then a random index in that chromosome. The value at that index of the chromosome is then altered to a different number within the range defined by the other numbers on either side. An example of this is shown in Table 3.

Table 3 Example mutation (notice the changed B value in $\mu_{\mathtt{medium}}$)

	A	*B*	*C*	*D*
(a) Original individual				
$\mu_{\mathtt{low}}$	0.33	0.5	0.66	1
$\mu_{\mathtt{medium}}$	0.33	*0.5*	0.66	1
$\mu_{\mathtt{high}}$	0.33	0.5	0.66	1
(b) Mutant (mutated B value in μ_{medium})				
	A	*B*	*C*	*D*
$\mu_{\mathtt{low}}$	0.33	0.5	0.66	1
$\mu_{\mathtt{medium}}$	0.33	*0.45*	0.66	1
$\mu_{\mathtt{high}}$	0.33	0.5	0.66	1

3.2.6 Termination

Since the optimal solution is not known *a priori*, the MBGM termination criterion is used (Martí et al., 2007). This methodology tracks the number of generations since the last time a new Pareto front was created. If at any point in time, more generations have passed than twice as many as the maximum number of generations between the creation of a new Pareto front, then the MOGA is labelled as having reached a steady state and unlikely to yield further optimisations. The algorithm therefore terminates, returning the Pareto front of the last generation as the result of evolution.

3.2.7 Result

The result of the optimisation is a Pareto front of individuals with non-dominated fitness values. The actual values seen in this Pareto for each of the three ports of interest are discussed in Section 4.

3.3 Evaluating data sources

Multiple independent data sources were found in the correlated data. Two of these data sources each contained a comparably large fraction of the combined data, making them fitting candidates for this study. The distribution of records per data source is specified in Table 10 and illustrated in Figure 7. Data from these two sources were extracted from the dataset to form two smaller, independent datasets upon which the optimisation was rerun. The results of the running the optimisation on these datasets are discussed in Section 4.

4 Results

The results from deploying the optimisation on the dataset for three ports of interest are presented in this section. As well, the results of evaluating the optimality of the two data sources mentioned in Subsection 3.3 are presented here.

4.1 Vessel service optimisation

Thirty independent runs of the optimisation were performed to collect statistically significant data. The mean and 95% confidence intervals are reported in this section

The vessel service optimisation was performed on three separate ports of interest, namely Port of Montreal [from a previous study (Panchapakesan et al., 2019)], Port of Halifax, and Port of Hong Kong. Previous studies have shown that at the Port of Montreal, the optimisation yields significant performance improvements in vessel service time.

4.1.1 Port of Montreal

The results of the optimisation performed on the Port of Halifax are shown in Table 4. The characteristics of the evolved fuzzy system that yielded this performance are shown in Table 5 and the fuzzy system it describes is illustrated in Figure 2.

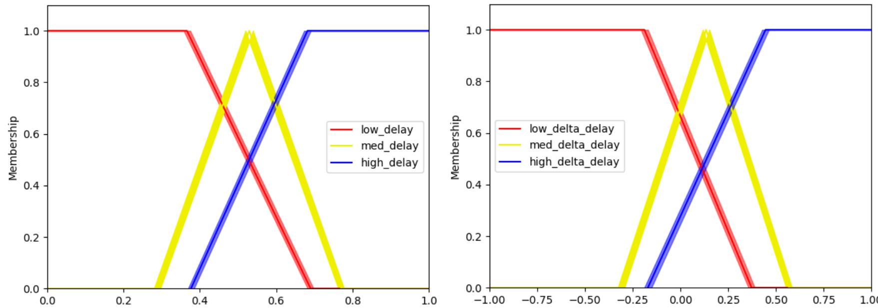
Table 4 Performance of evolved fuzzy systems for Port of Montreal

	Mined, real-world performance	*Optimised performance*
Number of crews used	63	23.654 ± 0.05
Total service time	77 days, 12 hours, 51 min, and 55 sec.	4 days, 4 hours, 27 min, 53 sec

Table 5 Mean characteristics of evolved fuzzy systems for Port of Montreal

Delay	μ_{low}	μ_{medium}	μ_{high}
A	0	0	0.377 ± 0.008
B	0	0.289 ± 0.008	0.682 ± 0.009
C	0.366 ± 0.009	0.529 ± 0.01	1
D	0.690 ± 0.008	0.771 ± 0.007	1
Δ *delay*	μ_{low}	μ_{medium}	μ_{high}
A	–1	–1	–0.170 ± 0.019
B	–1	–0.311 ± 0.017	0.446 ± 0.017
C	–0.188 ± 0.018	0.135 ± 0.017	1
D	0.373 ± 0.016	0.570 ± 0.015	1

Figure 2 Mean fuzzy membership functions for Port of Montreal (see online version for colours)



4.1.2 Port of Halifax

The results of the optimisation performed on the Port of Montreal are shown in Table 6. The characteristics of the evolved fuzzy system that yielded this performance are shown in Table 7 and the fuzzy system it describes is illustrated in Figure 3. A sample of the optimised vessel schedule is shown in Figure 4.

Table 6 Performance of evolved fuzzy systems for Port of Halifax

	Mined, real-world performance	*Optimised performance*
Number of crews used	33	3
Total service time	12 days, 11 hours, 50 minutes, 41 seconds	9 hours, 17 minutes, 54 seconds

Table 7 Mean characteristics of evolved fuzzy systems for Port of Halifax

Delay	μ_{low}	μ_{medium}	μ_{high}
A	0	0	0.4 ± 0.009
B	0	0.32 ± 0.008	0.72 ± 0.007
C	0.29 ± 0.009	0.56 ± 0.009	1
D	0.68 ± 0.009	0.76 ± 0.007	1
Δ *delay*	μ_{low}	μ_{medium}	μ_{high}
A	–1.0	–1.0	–0.2 ± 0.019
B	–1.0	–0.31 ± 0.018	0.38 ± 0.017
C	–0.33 ± 0.018	0.09 ± 0.017	1.0
D	0.35 ± 0.16	0.54 ± 0.014	1.0

Figure 3 Mean fuzzy membership functions for Port of Halifax (see online version for colours)

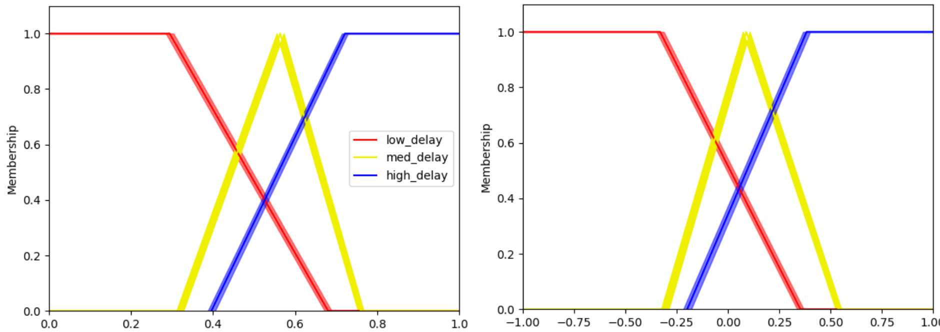
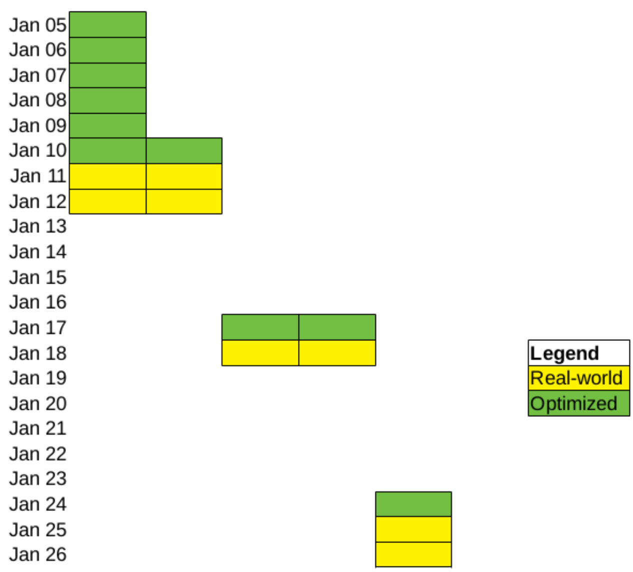


Figure 4 Optimised service schedule for the Port of Halifax (see online version for colours)



4.1.3 Port of Hong Kong

The results of the optimisation performed on the Port of Hong Kong are shown in Table 8. The characteristics of the evolved fuzzy system that yielded this performance are shown in Table 9 and the fuzzy system it describes is illustrated in Figure 5. A sample of the optimised vessel schedule is shown in Figure 6.

Table 8 Performance of evolved fuzzy systems for Port of Hong Kong

	Mined, real-world performance	*Optimized performance*
Number of crews used	88	23.097 ± 0.03
Total service time	73 days, 10 hours, 57 hours, 7 seconds	1 day, 15 hours, 18 minutes, 48 seconds

Table 9 Mean characteristics of evolved fuzzy systems for port of Hong Kong

Delay	μ_{low}	μ_{medium}	μ_{high}
A	0	0	0.35 ± 0.008
B	0	0.29 ± 0.007	0.67 ± 0.009
C	0.34 ± 0.01	0.53 ± 0.008	1
D	0.67 ± 0.008	0.75 ± 0.007	1
Δ delay	μ_{low}	μ_{medium}	μ_{high}
A	–1.0	–1.0	–0.27 ± 0.017
B	–1.0	–0.39 ± 0.017	0.37 ± 0.016
C	–0.36 ± 0.017	0.09 ± 0.017	1.0
D	0.30 ± 0.018	0.48 ± 0.017	1.0

Table 10 Distribution of records by data source

Source	*Number of records*
Source 1	4,048,309
Source 2	2,243
Source 3	408
Source 4	12,051
Source 5	3,932,275
Source 6	1

Figure 5 Mean fuzzy membership functions for Port of Hong Kong (see online version for colours)

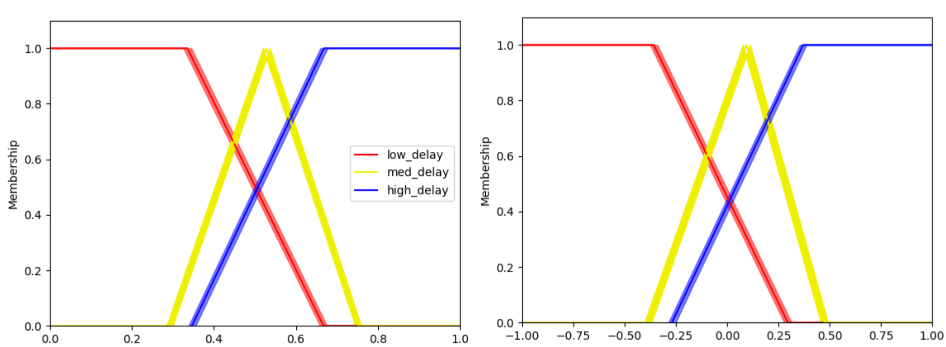
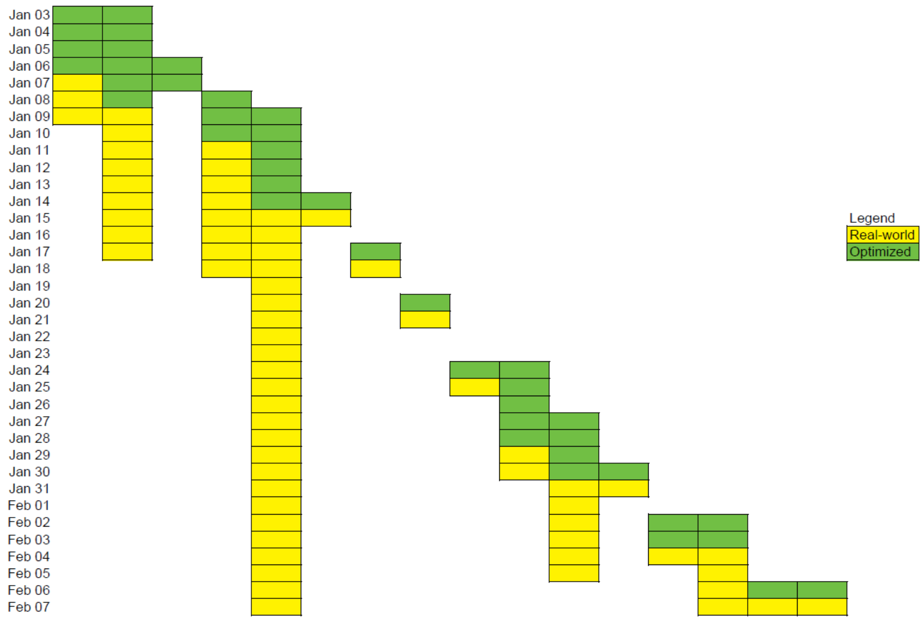


Figure 6 Optimised service schedule for the Port of Hong Kong (see online version for colours)



4.2 Data source selection

In order to meaningfully select between data sources, each data source was evaluated for its quality. The optimality of a data source is determined by training a model on data exclusively from this source and testing the performance of the trained model. The training and testing were performed in the same methodology as described in Subsection 3.2. In order to create the datasets for each data source, the original dataset was split into six sets, each one corresponding to one of the six AIS data sources present in the original dataset. The number of records from each data source is shown in Figure 7. Since sources 1 and 5 each have approximately 50% of all the records in the entire dataset, they were used for this analysis. On the other hand, since sources 2, 3, 4, and 6 contained a negligible number of AIS records, they did not provide sufficiently many records to perform this analysis and were preemptively filtered out.

With the separated data from sources 1 and 5, three datasets were created. The first contained data from only source 1, the second from only source 5, and the last from both sources 1 and 5. Next, three models were evolved using the same methodology as described in Subsection 3.2, one using each of the three new datasets. The resulting models were then run to determine the two-dimensional fitness values of their respective performances, so that they could be compared. The quality of each data source is measured as the optimality of the model resulting from training on the corresponding dataset. Since each data source accounts for approximately half the data in the original dataset, the equivalent of a model trained on the original dataset is expected to complete vessel service in approximately half the time with approximately half the resource deployment. This is therefore the benchmark against which models trained on either dataset will be compared. This expectation is shown in Table 11.

Figure 7 Distribution of records by data source (see online version for colours)

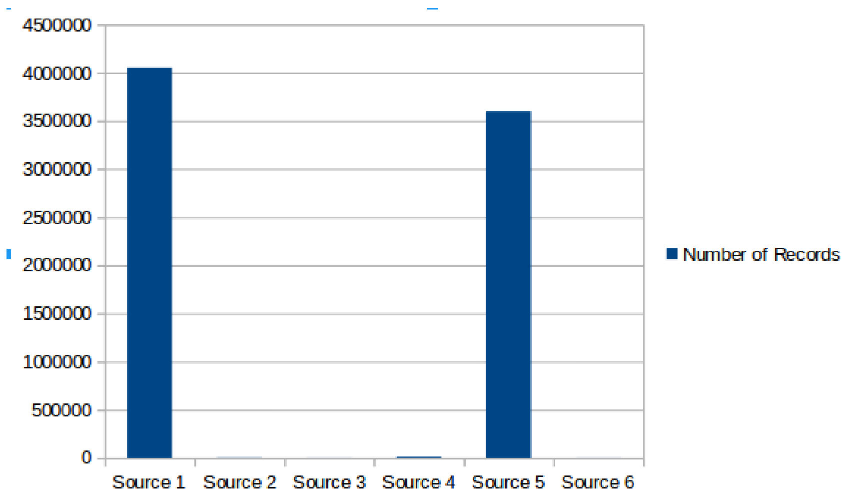


Table 11 Optimiser performance expectation

	(a) Port of Montreal	
	Optimised performance (with all data sources)	*Optimiser expectation (with source 1 or 5)*
Number of crews used	23.654 $\pm$ 0.05	11.827 $\pm$ 0.03
Total service time	4 days, 4 hours, 27 min, 53 sec	2 days, 2 hours, 13 min, 56 sec
	(b) Port of Halifax	
	Optimised performance (with all data sources)	*Optimiser expectation (with source 1 or 5)*
Number of crews used	3	2
Total service time	9 hours, 17 minutes, 54 seconds	4 hours, 38 minutes, 57 seconds
	(c) Port of Hong Kong	
	Optimised performance (with all data sources)	*Optimizer expectation (with source 1 or 5)*
Number of crews used	23.097 $\pm$ 0.03	11.549 $\pm$ 0.02
Total service time	1 day, 15 hours, 18 minutes, 48 seconds	19 hours, 39 minutes, 24 seconds

In contrast to the expectations laid out in Table 11, the results of the optimisation performed on these datasets is shown in Table 12. The results show that source 1 is of better quality than source 5 since the fitness of the optimiser trained with data from it dominates the fitness of the optimiser trained on data from source 5. However, considering that source 1 describes fewer vessel service instances than source 5 (see Table 13), it is possible to attribute this performance improvement to sample bias. Investigating this would require a larger dataset from sources 1 and 5, which are left as areas of further study that are out of the scope of the current work.

Table 12 Optimisation results on split datasets

(a) Optimisation results		
Data source	*Number of crews used*	*Total service time*
Real-world performance	63	77 days, 12 hours, 51 min, 55 sec.
Optimisation using all data sources	23.654 ± 0.05	4 days, 4 hours, 27 min, 53 sec
Optimisation using data source 1	9.26 ± 0.03	1 day, 8 hours, 15 min, 41 sec
Optimisation using data source 5	12.27 ± 0.04	1 day, 16 hours, 46 min, 54 sec
Optimisation using data sources 1 and 5	22.367 ± 0.05	4 days, 3 hours, 14 min, 56 sec
(b) Comparison to expectations		
Data source	*Improvement on crew usage*	*Improvement on total service time*
Optimisation using data source 1	21.7%	35.78%
Optimisation using data source 5	–3.8%	18.81%
Optimisation using data sources 1 and 5	5.4%	1.21%

Table 13 Vessel services per dataset

Data source	*Number of usable vessel services to optimise*
All data sources	27
Data source 1	15
Data source 5	22
Data sources 1 and 5	27

Sample bias in source 1 aside, the results show an improvement in the trained model when using data from both sources 1 and 5. This improvement confirms that excluding sources 2, 3, 4, 6 from the training data does improve the performance of the optimiser, supporting the arguments for data source selection within the original dataset. This combined optimiser does not dominate over the optimiser trained on source 5. Yet, it only improves upon the set benchmark (unlike its counterpart), which further supports the inclusion of data from source 1 when using source 5.

4.3 Optimiser robustness

In an attempt to saturate the resource utilisation, a new dataset was created for the Port of Halifax and Port of Hong Kong. This dataset was synthetically imputed to contain twice as many voyages as the original dataset. This was accomplished by creating a duplicate, 'twin' voyage for each voyage seen in the original dataset. The identifying information for the vessel and the voyage (namely the voyage ID and the vessel MMSI and IMO) for these twin voyages were randomly reassigned to unique values not seen in the original dataset, so as to double the vessel traffic as realistically as possible. Additionally, in order to resolve any conflicts due to multiple vessels sharing the same geo-spatial and temporal coordinates, the timestamps on the imputed contacts were offset by 30 minutes. An example of these changes is shown in Table 14. Note that prefixing '100' maintains the uniqueness over all MMSIs, IMOs, and voyage IDs, all of which were guaranteed to be unique in the original dataset.

Table 14 Creating a new AIS contact

Field	*Original contact*	*Synthetic contact*
MMSI	112358	**100**112358
IMO	99342	**100**99342
Voyage ID	92356ea3-d442-4e01-af8e-ddfae4bf68dc	**100**-92356ea3-d442-4e01-af8e-ddfae4bf68dc
Timestamp	14 May 2019 17:25:00	14 May 2019 17:55:00

Since each draught value in the original dataset is now present twice in the imputed datset, the container load through the port is effectively doubled. The same optimisation was run on this 'doubled' dataset and the resultant fuzzy membership functions are shown in this section.

Table 15 Performance of evolved fuzzy systems on doubled load at Port of Halifax

	Mined, real-world performance	*Optimised performance on real-world service load*	*Performance on doubled service load*
Number of crews used	33	3	3
Total service time	12 days, 11 hours, 50 minutes, 41 seconds	9 hours, 17 minutes, 54 seconds	9 hours, 17 minutes, 54 seconds

The results show that the evolved fuzzy system is able to easily handle double the service throughput without any significant additional resource requirements, making this a very robust optimised solution. There is a noted insignificant increase in the resource deployment at Port of Hong Kong. However, since both the real-world load and the synthetically doubled load, both require 24 cranes, this difference is considered negligible. The limits of such an optimised resource deployment model are difficult to identify, without additional data and are therefore left as future directions of investigation.

Table 16 Mean characteristics of evolved fuzzy systems for Port of Halifax

Delay	μ_{low}	μ_{medium}	μ_{high}
A	0	0	0.443 ± 0.009
B	0	0.349 ± 0.008	0.738 ± 0.007
C	0.358 ± 0.008	0.580 ± 0.008	1.0
D	0.690 ± 0.007	0.803 ± 0.006	1.0
Δ *delay*	μ_{low}	μ_{medium}	μ_{high}
A	–1.0	–1.0	–0.269 ± 0.015
B	–1.0	–0.316 ± 0.016	0.387 ± 0.016
C	–0.309 ± 0.017	0.063 ± 0.015	1.0
D	0.444 ± 0.016	0.492 ± 0.013	1.0

Figure 8 Mean fuzzy membership functions for Port of Halifax with doubled traffic (see online version for colours)

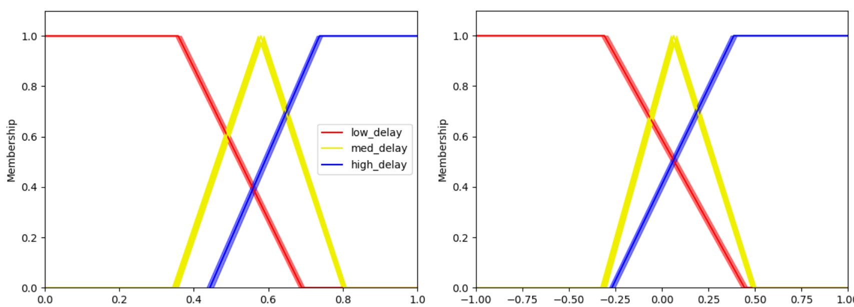


Figure 9 Mean fuzzy membership functions for Port of Hong Kong with doubled traffic (see online versionfor colours)

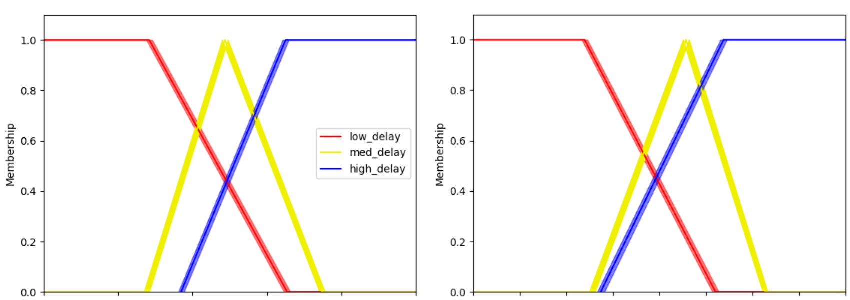


Table 17 Performance of evolved fuzzy systems on doubled load for Port of Hong Kong

	Mined, real-world performance	*Optimised performance*	*Performance on doubled service load*
Number of crews used	88	23.097 ± 0.03	23.173 ± 0.04
Total service time	73 days, 10 hours, 57 hours, 7 seconds	1 day, 15 hours, 18 minutes, 48 seconds	1 day, 15 hours, 18 minutes, 48 seconds

Table 18 Mean characteristics of evolved fuzzy systems for Port of Hong Kong with doubled traffic

Delay	μ_{low}	μ_{medium}	μ_{high}
A	0	0	0.369 ± 0.008
B	0	0.276 ± 0.007	0.650 ± 0.008
C	0.284 ± 0.008	0.487 ± 0.008	1
D	0.654 ± 0.008	0.748 ± 0.007	1
Δ *delay*	μ_{low}	μ_{medium}	μ_{high}
A	–1.0	–1.0	–0.315 ± 0.019
B	–1.0	–0.362 ± 0.015	0.345 ± 0.017
C	–0.403 ± 0.015	0.143 ± 0.016	1.0
D	0.302 ± 0.017	0.568 ± 0.014	1.0

5 Conclusions and future work

It has been shown that a MOGA can optimise a fuzzy system to control the adaptive deployment of port-side resources. It has also been shown that separating multiple data sources and selectively using the data from a subset thereof yields superior optimisation results. Some optimisation results call to question whether sample bias may have occurred, which can be dispelled with more data than was available for this study. Further, pruning some data sources has been shown to improve the performance of the resulting optimiser, which can be added to the data ingestion pipeline. Additionally, other measures for the *a priori* evaluation of a dataset could help better predict the quality of the dataset. Such measures have been proposed in Falcon et al. (2014) and could be further evaluated in the context of this work.

The resource deployment model is also able to handle twice the container load seen in the real world without altering the resource deployment, making it a very robust model with higher capacity than originally designed for. With more data, the model can be pushed further to determine its true capacity.

In an expansion of previous work (Panchapakesan et al., 2019), this resource optimisation model has been shown to generalise to multiple commercial maritime ports across multiple time periods, and against twice the known service load. This is in contrast to other models that require *a priori* knowledge of incoming vessel schedules in order to optimise resource deployment. This model is therefore *online* and is able to maintain optimal performance when faced with vessel delays, etc. unlike previously proposed models.

Further study is warranted into the optimisation of the fuzzy rule base and the specification of the defuzzifier. As well, collecting more data from various sources can help better inform the usefulness of the data from each source.

Acknowledgements

We would like to thank:

1 Southern Ontario Smart Computing Innovation Platform (SOSCIP) and SciNet on whose infrastructure our experiments were run.[1]

2 Alyssa Fred Wai-Yi Wong, a professional highway truck driver, for sharing her expertise used in data creation and validation.

3 Ontario Centers of Excellence (OCE) and NSERC for their CRD grant for 'Big data analytics for the maritime internet of things (IoT)', through which this study was funded.

4 Larus Technologies for sharing their domain expertise in this work.

References

Abualhaol, I., Falcon, R., Abielmona, R. and Petriu, E. (2018) 'Data-driven vessel service time forecasting using long short-term memory recurrent neural networks', in *2018 IEEE International Conference on Big Data (Big Data)*, IEEE, pp.2580–2590.

Blasch, E., Bossé, É. and Lambert, D.A. (2012) *High-level Information Fusion Management and Systems Design*, Artech House, Boston, USA.

Cheng, D., Rao, J., Jiang, C. and Zhou, X. (2015) 'Resource and deadline-aware job scheduling in dynamic Hadoop clusters', in *2015 IEEE International Parallel and Distributed Processing Symposium*, IEEE, pp.956–965.

Cheraghchi, F., Abualhaol, I., Falcon, R., Abielmona, R., Raahemi, B. and Petriu, E. (2017) 'Big-data-enabled modelling and optimization of granular speed-based vessel schedule recovery problem', in *2017 IEEE International Conference on Big Data (Big Data)*, pp.1786–1794, IEEE.

Davis, L. (1991) *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York.

Falcon, R., Abielmona, R., Billings, S., Plachkov, A. and Abbass, H. (2014) 'Risk management with hard-soft data fusion in maritime domain awareness', in the *2014 Seventh IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, pp.1–8, IEEE.

Jamshidi, P., Sharifloo, A., Pahl, C., Arabnejad, H., Metzger, A. and Estrada, G. (2016) 'Fuzzy self-learning controllers for elasticity management in dynamic cloud architectures', in *2016 12th International ACM SIGSOFT Conference on Quality of Software Architectures (QoSA)*, IEEE, pp.70–79.

Li, Z., Stan, L., Czaplewski, D.A., Yang, X. and Gao, J. (2019) 'Broadband infrared binary-pattern metasurface absorbers with micro-genetic algorithm optimization', *Optics Letters*, Vol. 44, No. 1, pp.114–117.

Martí, L., García, J., Berlanga, A. and Molina, J.M. (2007) 'A cumulative evidential stopping criterion for multiobjective optimization evolutionary algorithms', in *Proceedings of the 9th Annual Conference Companion on Genetic and Evolutionary Computation*, ACM, pp.2835–2842.

Panchapakesan, A., Abielmona, R. and Petriu, E. (2019) 'Optimizing maritime vessel service time with adaptive quay crane deployment through level 4 hard-soft information fusion', in *Proc. of the 22nd International Conference on Information Fusion*, ISIF, pp.1–10.

Perez, H.M., Chang, R., Billings, R. and Kosub, T.L. (2009) 'Automatic identification systems (AIS) data use in marine vessel emission estimation', in *18th Annual International Emission Inventory Conference*, Vol. 14, p.e17.

Peterkofsky, R.I. and Daganzo, C.F. (1990) 'A branch and bound solution method for the crane scheduling problem', *Transportation Research Part B: Methodological*, Vol. 24, No. 3, pp.159–172.

Rey Horn, J., Nafpliotis, N. and Goldberg, D.E. (1994) 'A niched pareto genetic algorithm for multiobjective optimization', in *Proceedings of the First IEEE conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, Citeseer, Vol. 1, pp.82–87.

Suri, S. and Vijay, R. (2019) 'A bi-objective genetic algorithm optimization of chaos-dna based hybrid approach', *Journal of Intelligent Systems*, Vol. 28, No. 2, pp.333–346.

Teske, A., Falcon, R., Abielmona, R. and Petriu, E. (2017) 'Automating maritime risk assessment with genetic fuzzy systems', in *Proc. 2nd Int. Symp. Fuzzy Rough Sets (ISFUROS)*, pp.1–10.

Wang, L., Xu, J., Zhao, M., Tu, Y. and Fortes, J.A. (2011) 'Fuzzy modeling based resource management for virtualized database systems', in *2011 IEEE 19th Annual International Symposium on Modelling, Analysis, and Simulation of Computer and Telecommunication Systems*, IEEE, pp.32–42.

Wesolkowski, S., Francetic, N. and Grant, S.C. (2014) 'TraDE: training device selection via multi-objective optimization', *2014 IEEE Congress on Evolutionary Computation (CEC)*, pp.2617–2624.

Xu, Z-D., Huang, X-H., Xu, F-H. and Yuan, J. (2019) 'Parameters optimization of vibration isolation and mitigation system for precision platforms using non-dominated sorting genetic algorithm', *Mechanical Systems and Signal Processing*, Vol. 128, pp.191–201, DOI: https://doi.org/10.1016/j.ymssp.2019.03.031.

Zhang, C., Wan, Y-W., Liu, J. and Linn, R.J. (2002) 'Dynamic crane deployment in container storage yards', *Transportation Research Part B: Methodological*, Vol. 36, No. 6, pp.537–555 [online] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.463.6610&rep=rep1&type=pdf.

Zhang, C., Wan, Y-w., Liu, J. and Linn, R.J. (2002) 'Dynamic crane deployment in container storage yards', *Transportation Research Part B: Methodological*, Vol. 36, No. 6, pp.537–555.

Notes

1 Each GPC node runs Intel's Xeon E5540 8-core CPU at 2.53 GHz, with 16 GB RAM; each P7 node runs an IBM Power 755 server with four 8-core 3.3 GHz Power7 CPUs and 128 GB RAM. Detailed specifications can be found at https://www.scinethpc.ca/.