The Effects of Chromosome Length on Hyper-Heuristics for Solving the Maritime Inventory Routing Problems

Nurhadi Siswanto
Department of Industrial Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, 60111, Indonesia
siswanto@ie.its.ac.id

Abstract

This paper describes the effects of chromosome length for solving maritime inventory routing problems (MIRP) by using a hyper-heuristics based Genetic Algorithm (GA). The approach uses a set of heuristic combinations, each of which consist of strategies that correspond to a ship assignment. These strategies are represented by a chromosome that may have several assignments. We examine several number of chromosome length to encourage the evolution of good heuristics combinations. Moreover, a variation of several number chromosome length is necessary since we do not know in advance how many ship assignments are needed to cover demands during a predefined planning horizon. At every iteration a number of chromosomes are evaluated and evolved within a GA framework. In this study, the approach has been applied on several test cases for transporting multiple oil products from a production facility to some consumption ports, by using several heterogeneous ships with undedicated compartments. The results show that a hyper-heuristics based GA reaches the same global optimal as the solutions in the mathematical model, but with a significant decrease in computational time. Moreover, the use two numbers of chromosome length proves that three assignments in one step (3AiOS) mostly got better solutions and lower minimum total number of assignments than the two assignments (2AiOS).

Keywords
Hyper-heuristics, Genetic Algorithm, Chromosome Lenght, Maritime Transportation, Inventory Routing Problem.

1. Introduction

Maritime inventory routing problem (mIRP) can be defined as an inventory routing problem (IRP) that uses ships to distribute product(s). Many papers have discussed this problem, for example the model considered in Christiansen, and Nygreen (1998a and 1998b), Christiansen (1999), Hwang (2005) and Al-Khayyal and Hwang (2007). The first three papers considers a single product model, while the last two extends the problem to have multiple non-intermixable products. An extensive survey on maritime scheduling research is provided in Christiansen et al. (2007) and Christiansen and Fagerholt (2008).

Several papers show a real world application of mIRP, such as Christiansen et al. (2011), Furman et al. (2011) and Song and Furman (2013). The first paper discusses the distribution of multiple grade cements from two production ports to 28 consumption ports in Norway. In this problem, the delivery of a ship is limited to a pair of one production and one consumption port. They developed a method that consists of two components: a heuristic construction and a genetic algorithm to solve the problem. While the last two papers discussed the same problem, transporting a single vacuum gas oil (VGO) product from several production ports in Europe to refineries in the United States. In this problem, the ships may pick-up the product at multiple nodes before distributing it to the several ports. They use a method that consists of a large neighbourhood search and a branch-and-cut algorithm for solving this problem.

This paper also discusses a mIRP that delivers multiple products from a production port to several ports. The problem is similar to the model discussed in Christiansen et al. (2011). The similarities are multi-intermixable products, the undedicated compartments that need an algorithm to assign products into compartments and the continuous time model. However, in our model we assumed that the ships may visit up to two ports in one assignment, instead of one port in Christiansen et al. (2011). Because of this difference, we have a procedure to incorporate the sequence of visiting port and how to divide the product quantities between these two consumption ports. In contrast to our model, Furman et al. (2011) and Song and Furman (2013) discussed a single product and a discrete time model. However, they have a variation to have multiple pick-up and deliveries in their problem. Each problem in the mIRP has some uniqueness and the problem is of how to model the problem such that it is flexible enough to implement various scenarios of the problems as mentioned in Song and Furman (2013).

Heuristic has become a popular tool for solving various scheduling problems because of its superiority in solving problems within a reasonable time. However, as stated by Cowling et al. (2002) and Chakhlevitch and Cowling (2008), a heuristic may perform well for a certain problem instance, but it is unlikely that the heuristic may be applied successfully to a different instance of the problem. This problem specific method means that it is hard to implement it for a problem that has many variations, such as mIRP. Many researchers have proposed to use indirect solution heuristics to overcome this limitation. Cowling (2000) has introduced a concept of hyper-heuristics for this indirect heuristics and has applied the method in several scheduling problems as described in Cowling et al. (2000, 2002). The method uses heuristics to choose a lower level of heuristics. An extensive survey of the development of this method can be seen in Cowling et al. (2002).

This paper proposes to use an indirect method heuristic referred to hyper-heuristics as defined in Cowling et al. (2000). As far as we know, our approach is the first to use this approach in solving a maritime inventory routing problem. Although Christiansen et al. (2011) also used GA to solve mIRP, however their method is different from us. They develop a combination of a construction heuristic and a genetic algorithm. The heuristic is an iterative procedure to get a solution, however the solution given depends on the parameters generated in the GA. They uses GA to modify the parameters in the objective function to form different scenarios. Their domain is in the solution space, while ours is in the solution method.

Moreover, our method adds another feature that is an adaptive length of chromosome. As mentioned in Han et al. (2004), the purpose of this adaptation is to encourage the evolution of good heuristics combinations. Furthermore, the adaptive length is necessary in our problem since we do not know in advance how many ship assignments are needed to cover demands during the predefined planning horizon.

Han et al. (2004) uses an adaptive length of chromosome to solve a trainer scheduling problem. The evolution of good heuristics combinations can be achieved by removing poor performing heuristics from a chromosome and by also injecting others. The method selects two points, described as gene m and gene n, and evaluates whether the genes between these two points will improve the objective function or not. If it is improving, the block genes will be injected in the chromosome. Otherwise, they will be removed.

Although we also use an adaptive length of chromosome, we have a different approach because our problem is multi period problem. A decision at a certain time or stage will change the state or condition of ships or ports, for example the position of ships and their compartment levels, and the storage levels of the ports. If different decision is made at that particular time, the state or condition will be different from the previous one. Each decision will have its own effect. Because of that, the remove-inject method cannot be applied in our problem. The removal of one or more heuristics will make the initial state for the next heuristic gene, let us say gene n+1, different from the state before the genes are removed. In our approach, the genes of a chromosome can only be evaluated consecutively from the first gene until the last gene.

This paper is an extension to our previous paper as described in Siswanto et al. (in press). By using the same problem which is applied in a complex distribution network faced by a national oil company in Indonesia, the objective of this paper is to compare the performances of two differences number of chromosome length: two or three assignments in one step (2AiOS or 3AiOS). Since we do not describe the problem again, the reader may refer to our previous paper (Siswanto et al., in press) to have detailed problem description. The outline of this paper is as follows: section 2 describes the terminology and method of the hyper-heuristics based genetic algorithm. Then, section 3 reports the computational results of the case problems. Finally, the last section gives some concluding remarks.

2. Hyper-heuristics based Genetic Algorithm

In this section, we describe some terminologies used in the hyper-heuristics based Genetic Algorithm (GA) and its adaptive length first.

The first terminology is *an assignment*, which is defined as one movement of a ship from its current position to another port to deliver products. The movement may involve visiting one or two ports in between. The ship first visits a production port before going to consumption port(s). In this case, we enforce a limit of a maximum of two consumption ports that can be visited in one assignment. We define *a step* as the number of assignments in one chromosome. This can be seen as the number of steps of look ahead. Each assignment will change the condition or status of a collection of variables defined as *a state*, for example a ship's time and position, its compartment levels and their contents, and a port's time and storage levels. Ship's time is the time of a ship when it finishes its assignment, and hence when it is ready for the next assignment, while the port's time is the last time of a port visited by a ship. These two values are updated whenever a ship has completed its assignment. We denote SI_n as an initial state of assignment *n*, while SA_n is a state after processing assignment *n*. Obviously, $SA_n = SI_{n+1}$.

Next, *coverage day* (*CD*) is defined as the number of days that port *i* can cover the demand of product *k* from its storage before it runs out (for a discharge port) or before its storage reaches the maximum capacity (for a supply port). This parameter is used to determine which port and its storage needs to be served next. The less number of coverage days, the more prioritize a port to be visited by a ship. Hwang (2005) and Savelsbergh and Song (2007) use urgency to state coverage day.

In solving this maritime inventory routing problem, we consider hyper-heuristics as a set of heuristic and select one of the heuristics for use in each sub-problem. We identified there are four sub-problem in this problem. The first is ship selection rule to determine which ship to be assigned. The second is routing determination rule to select which ports need to be served by the selected ship. The third is the loading activity rule to determine the type and the quantity of products to be loaded and simultaneously to assign which product are to be loaded into which compartment. The last is the unloading rule to determine the type and the quantity of products to be unloaded.

In this hyper-heuristics approach, we consider a combination of heuristic that consist of four rules, i.e. one for each sub-problem. For example a combination of rules: S_0–R_0–L_0–U_0 means that a ship selected based on the least ship's time (S_0), the routing of the selected ship will be a production port and a port that has the least CD (R_0), we assign product [1] into compartment [1] and product [2] into compartment [2] of the selected ship with the quantity loaded as the compartment capacity (L_0), and unloaded all the products in the ship at the selected discharge port (U_0). This combination of heuristic can solve problems, but the performance of the solution may be different from other combinations. The discussion of single combination results can be seen in Siswanto et al. (2011). This combination of heuristics is a representation of an assignment of a ship as previously defined and will be encoded as a chromosome. The complete single heuristic for each sub-problem and how these hyper-heuristics are encoded as chromosome are explained in Siswanto et al. (in press).

In our method, the length of a chromosome depends on the number of assignments defined as one *step*. The previous combination S_0–R_0–L_0–U_0 can be an example one assignment in one step chromosome, called as 1AiOS chromosome. Another example, S_0–R_0–L_0–U_0–S_1–R_2–L_0–U_2 as a two assignments in one step (2AiOS) chromosome. The number of assignments in one step chromosome can be set to alter the solution space. The larger the number of assignments the larger the solution space which promote to find many good combinations. Although we can set as many as number of assignments in one step, let say three number of assignments in one step (3AiOS), the last (the third) assignment may not be processed due to a complete satisfied or a worse solution has been found. In this algorithm, a variable *sa* is used to specify the stopping assignment number.

It is also possible that another step is needed whenever the demand for the entire planning horizon has not been reached yet when the last assignment of a step is completed. This is why we called this chromosome as an adaptive because of the flexibility of the lenght of chromosome. From this point on, we use notation $gene_{inm}$ to represent the gene number *m* of assignment *n* in step *i*. Also, 2AiOS-- and 3AiOS-- represent scenarios of two and three assignments in one step. The (--) indicates the length of planning horizon we modeled. We will discuss in the computational study section how the number of assignments in one step will have effects in solving problems in the next section.

3. Computational Results and Analysis

The approach described in the previous section is an extension to the model introduced in Siswanto et al. (2011) and Siswanto et al. (in press). Instead of running several individual heuristics, the proposed method tries to find the best solution in a single run. We generate test problems (TP) under various conditions to demonstrate the applicability of the proposed method. The detail description of the test problems can be found in Siswanto et al. (2011).

In the hyper-heuristics based GA, we randomly generated a population of a hundred chromosomes at each generation. Each of the chromosomes in the population consists of two or three assignments in one step. We set the maximum number of generation is 80 for each step, intensifying a step is set to be 10 and the maximum steps is set to be eight. The chromosomes are evaluated based on the least fitness function by using algorithm 1. We keep 20 percent of the best chromosomes from the old population to form a new population in each generation.

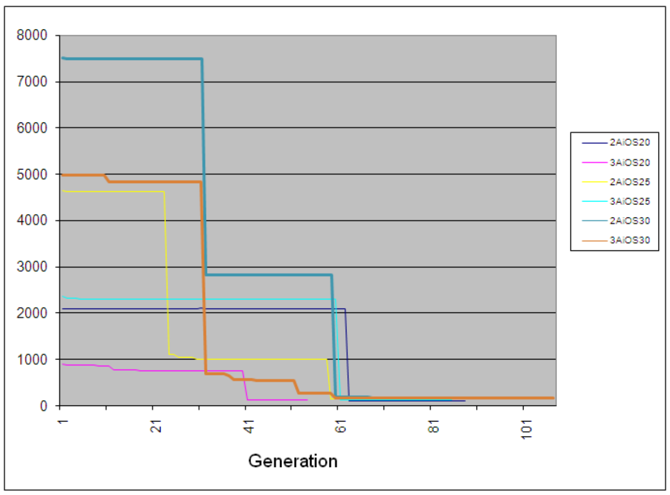

Figure 1. The Progress of Fitness Function of Various Scenarios of Test Problem 1 as a Function of Generation

We conducted experiments to compare between the number of assignments in one step, which are two and three assignments. Moreover, we extended to solve the test problems until 30 day planning horizon (PH) to demonstrate the applicability of an adaptive length chromosome. Figure 1, Figure 2, and Figure 3 represent a solution of various scenarios of test problem 1. These figures show the progress of fitness function, total cost and the least Coverage Day among all storages in the ports as a function of generation. At the beginning of period the minimum total cost is the lowest value, on the other hand the fitness function is the highest one at that time. This is caused by the highest completion cost, added to the total cost since the least Coverage Day is the lowest value at this time. When the number of step is added, the total cost and the least Coverage Day significantly increase, on the other hand the fitness function significantly decrease. All of these scenarios have two steps, except scenario 2AiOS30 that has three steps. When generation becomes larger, all the demands have been fulfilled and they reach their levels to satisfy the demands until the end of planning horizon. At this point, the total cost will have the same value with the fitness function because there is no late shipment.

As shown in Table 1, the hyper-heuristics got solutions for all test problems. The running time of the hyper-heuristics depends on the number of generation needed to solve the problem. The longer the number of day planning horizon the longer running time. Moreover, 2AiOS have shorter running time compared to 3AiOS, a significant decrease running time compared to mathematical model. Although the solution of the proposed method does not guarantee to always get a globally optimal solution as in the mathematical model, the best solutions of the proposed method have the same objective functions as the ones of mathematical model solutions. The proposed method provides good quality feasible solutions within an acceptable running time. Furthermore, almost all 3AiOS have the same or better objective functions compared to 2AiOS. Only problem test 1 of 20 day planning horizon and problem test 3 of 30 day planning horizon have 2AiOS better than 3AiOS. However, 2AiOS are relatively easy to converge compared to 3AiOS because of their lower standard deviations.

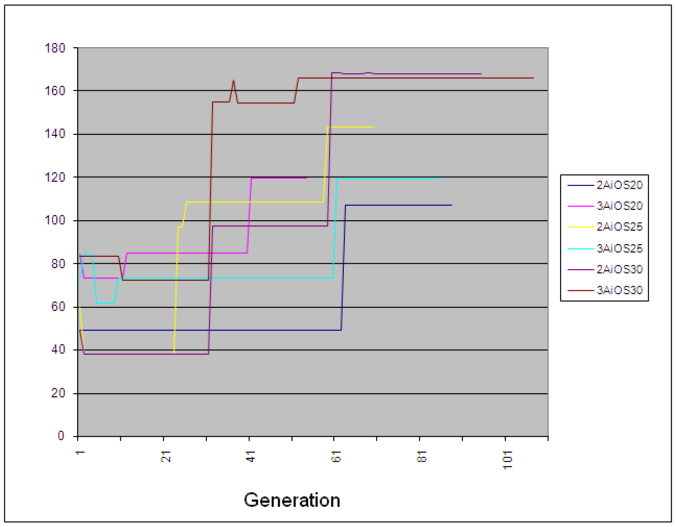

Figure 2. The Progress of Total Cost of Various Scenarios of Test Problem 1 as a Function of Generation

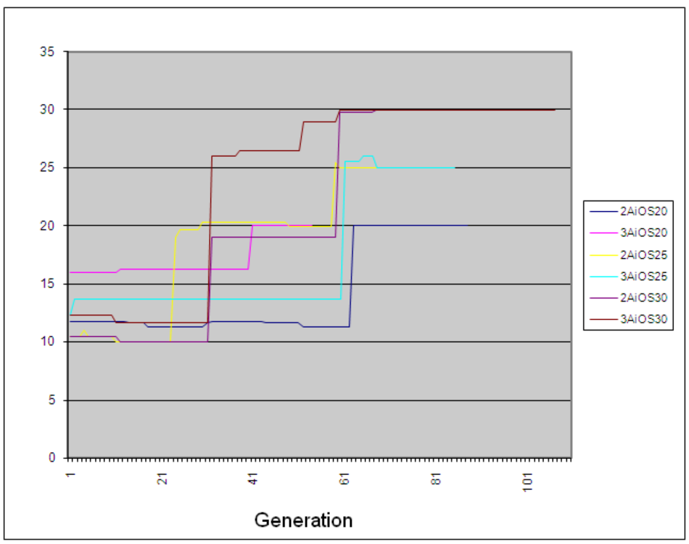

Figure 3. The Progress of the Least Coverage Day of Various Scenarios of Test Problem 1 as a Function of Generation

In this case, we only compared to the method to mathematical model as discussed in Siswanto et al. (2011), up to 15 day planning horizon. Beyond this planning horizon, the running times of mathematical model are beyond the time limit. As shown in the Table 1, there are no difference in the best solutions between 2AiOS and 3AiOS for all test problems with 10 and 15 day planning horizon compared to mathematical model.

Table 1. The Result of Hyper-Heuristics based GA with comparing to Mathematical Model

TP	PH	AiOS	Name of Scenario	Best Mathema-tical Model Solution	Hyper-Heuristics based GA (10 running repetition)					
					Best Solution	Gap (%)	Maximum	Average	Standard Deviation	Average Running Time (in second)
1	10	2	2AiOS10	38.0	38.0	0	38.0	38.0	0	291.6
		3	3AiOS10	38.0	38.0	0	38.0	38.0	0	352.1
	15	2	2AiOS15	73.0	96.0	31.5	96.0	96.0	0	800.1
		3	3AiOS15	73.0	73.0	0	73.0	73.0	0	807.7
	20	2	2AiOS20	-	105.0	-	105.0	105.0	0	910.7
		3	3AiOS20	-	120.0	-	120.0	120.0	0	1,514.3
	25	2	2AiOS25	-	143.5	-	143.5	143.5	0	1,228.0
		3	3AiOS25	-	119.5	-	119.5	119.5	0	2,113.8
	30	2	2AiOS30	-	168.0	-	168.0	168.0	0	1,570.3
		3	3AiOS30	-	166.0	-	177.0	168.4	4.56	2,829.6
2	10	2	2AiOS10	54.9	54.9	0	54.9	54.9	0	670.4
		3	3AiOS10	54.9	54.9	0	54.9	54.9	0	795.3
	15	2	2AiOS15	106.7	132.0	30.1	138.8	132.7	2.15	904.2
		3	3AiOS15	106.7	106.7	0	118.3	113.9	2.84	1,180.0
	20	2	2AiOS20	-	169.6	-	172.5	171.1	1.48	1,308.3
		3	3AiOS20	-	140.8	-	185.7	155.8	12.62	1,601.1
	25	2	2AiOS25	-	172.1	-	206.7	192.0	8.27	1,670.5
		3	3AiOS25	-	171.1	-	214.3	183.7	12.70	2,099.8
	30	2	2AiOS30	-	234.3	-	244.6	239.5	7.28	2,650.5
		3	3AiOS30	-	209.8	-	234.9	221.9	7.19	2,347.7
3	10	2	2AiOS10	42.1	42.1	0	42.1	42.1	0	216.9
		3	3AiOS10	42.1	42.1	0	42.1	42.1	0	269.6
	15	2	2AiOS15	76.8	84.5	10.0	84.5	84.5	0	684.6
		3	3AiOS15	76.8	76.8	0	85.5	91.4	2.17	932.4
	20	2	2AiOS20	-	134.3	-	134.3	134.3	0	1,305.5
		3	3AiOS20	-	110.4	-	115.8	111.5	2.28	1,989.0
	25	2	2AiOS25	-	146.8	-	146.8	146.8	0	1,221.1
		3	3AiOS25	-	140.8	-	174.4	155.9	13.83	1,904.0
	30	2	2AiOS30	-	186.5	-	199.6	190.7	6.19	1,518.9
		3	3AiOS30	-	199.4	-	202.5	201.2	0.82	1,971.6
4	10	2	2AiOS10	137.0	137.0	0	137.0	137.0	0	374.0
		3	3AiOS10	137.0	137.0	0	137.0	137.0	0	489.0
	15	2	2AiOS15	269.0	289.0	7.4	289.0	289.0	0	1,066.9
		3	3AiOS15	269.0	269.0	0	277.0	273.3	3.83	1,127.7
	20	2	2AiOS20	-	432.0	-	432.0	432.0	0	1,476.9
		3	3AiOS20	-	314.0	-	444.0	363.7	46.85	2,076.5
	25	2	2AiOS25	-	532.0	-	532.0	532.0	0	1,630.7
		3	3AiOS25	-	416.0	-	515.0	452.6	50.55	2,192.6
	30	2	2AiOS30	-	623.0	-	665.0	659.8	14.85	1,659.1
		3	3AiOS30	-	545.0	-	577.0	561.8	80.12	2,651.4
5	10	2	2AiOS10	91.0	91.0	0	91.0	91.0	0	226.1
		3	3AiOS10	91.0	91.0	0	91.0	91.0	0	271.3
	15	2	2AiOS15	208.0	208.0	0	216.0	209.6	3.85	493.0
		3	3AiOS15	208.0	208.0	0	208.0	208.0	0	644.8
	20	2	2AiOS20	-	378.0	-	378.0	378.0	0	1,402.3
		3	3AiOS20	-	304.0	-	314.0	305..0	3.16	1,865.2
	25	2	2AiOS25	-	406.0	-	406.0	406.0	0	1,043.9
		3	3AiOS25	-	346.0	-	457.0	391.0	52.21	1,861.1
	30	2	2AiOS30	-	488.0	-	607.0	523.5	42.09	1,563.2
		3	3AiOS30	-	482.0	-	584.0	528.8	36.06	2,539.9

Table 2 shows the length of chromosome needed to solve each of the scenarios. The table provides minimum, maximum and average of number of steps and total number of assignments for each of the scenarios. For example, scenario 2AiOS30 of test problem 1 needs 3 steps each of which has two assignments. While 3AiOS30 of the same problem needs only 2 steps each of which 3 has assignments. Furthermore, the last step may not use all the assignments provided as shown in 2AiOS15 and 3AiOS25 of test problem 1. In the first scenario, the total length of chromosome is five assignments: the first two steps have two assignments while the last step has only one of two assignments provided. In the second scenario, it has also five total assignments: one step with three assignments while the last step with only two of three assignments provided. Furthermore, all 3AiOS have the same or lower minimum total number of assignments compared to 2AiOS. The longer length chromosome encourages the evolution of good heuristics combinations. However it cost the longer running time. On the other hand, the average total numbers of assignments of 3AiOS30 are greater number than the ones of 2AiOS30 in Test problem 2, 3, and 5. Because some results of 3AiOS30 have total number of assignments greater than the ones of 2AiOS30 in those problems.

Table 2. The Length Chromosome Needed to Solve Each of the Scenarios

TP	PH	Number Assignments in One Step	Name of Scenario	Minimum		Maximum		Average Number of Steps	Average Total Number of Assignments
				Number of Steps	Total number of Assignments	Number of Steps	Total number of Assignments		
1	10	2	2AiOS10	1	2	1	2	1	2
		3	3AiOS10	1	2	1	2	1	2
	15	2	2AiOS15	2	3	3	5	2.1	3.3
		3	3AiOS15	1	3	1	3	1	3
	20	2	2AiOS20	2	4	2	4	2	4
		3	3AiOS20	2	4	2	4	2	4
	25	2	2AiOS25	3	6	3	6	3	6
		3	3AiOS25	2	5	2	5	2	5
	30	2	2AiOS30	3	6	3	6	3	6
		3	3AiOS30	2	6	2	6	2	6
2	10	2	2AiOS10	1	2	1	2	1	2
		3	3AiOS10	1	2	1	2	1	2
	15	2	2AiOS15	2	4	2	4	2	4
		3	3AiOS15	1	3	1	3	1	3
	20	2	2AiOS20	2	4	2	4	2	4
		3	3AiOS20	1	3	2	4	1.1	3.1
	25	2	2AiOS25	3	5	3	6	3	5.1
		3	3AiOS25	2	4	2	6	2	4.9
	30	2	2AiOS30	3	5	3	5	3	5
		3	3AiOS30	2	5	2	6	2	5.6
3	10	2	2AiOS10	1	1	1	1	1	1
		3	3AiOS10	1	1	1	1	1	1
	15	2	2AiOS15	1	2	1	2	1	2
		3	3AiOS15	1	2	1	2	1	2
	20	2	2AiOS20	2	3	2	3	2	3
		3	3AiOS20	1	3	1	3	1	3
	25	2	2AiOS25	2	4	2	4	2	4
		3	3AiOS25	1	3	2	4	1.5	3.5
	30	2	2AiOS30	2	4	2	4	2	4
		3	3AiOS30	2	4	2	5	2	4.4
4	10	2	2AiOS10	1	2	1	2	1	2
		3	3AiOS10	1	2	1	2	1	2
	15	2	2AiOS15	2	3	2	3	2	3
		3	3AiOS15	1	3	1	3	1	3
	20	2	2AiOS20	2	4	2	4	2	4
		3	3AiOS20	1	3	2	4	1.3	3.3
	25	2	2AiOS25	3	5	3	5	3	5
		3	3AiOS25	2	4	2	5	2	4.3
	30	2	2AiOS30	3	6	3	6	3	6
		3	3AiOS30	2	5	2	5	2	5
5	10	2	2AiOS10	1	1	1	1	1	1
		3	3AiOS10	1	1	1	1	1	1
	15	2	2AiOS15	1	2	1	2	1	2
		3	3AiOS15	1	2	1	2	1	2
	20	2	2AiOS20	2	3	2	3	2	3
		3	3AiOS20	1	3	1	3	1	3
	25	2	2AiOS25	2	4	2	4	2	4
		3	3AiOS25	1	3	2	4	1.4	3.4
	30	2	2AiOS30	2	4	2	4	2	4
		3	3AiOS30	2	4	2	5	2	4.1

4. Conclusion

In this paper, we show that a Hyper-Heuristics based GA reaches the same global optimal as the solutions in the mathematical model for solving shorter planning horizon (i.e. 10 and 15 day planning horizon), but with a significant decrease in computational time. We also demonstrated the use adaptive length chromosome to solve longer planning horizon. In this method, three assignments in one step (3AiOS--) mostly get better solutions and lower minimum total number of assignments than the two assignments (2AiOS--). However, they cost longer running time. These successful results indicates that the method can be applied to larger test problems (i.e. longer planning horizon or modeling more ports, ships or/and products to be served). Moreover, the flexibility to add strategies means the method can be further extended to adapt to the various features of the problems.

References

Al-Khayyal, F. and Hwang, S.J., Inventory constrained maritime routing and scheduling for multi-commodity liquid bulk, Part I: Applications and model. *European Journal of Operational Research*, vol. 176, no. 1, pp.106-130, 2007.

Chakhlevitch, K. and Cowling, P., Hyperheuristics: recent developments. In *Adaptive and multilevel metaheuristics* (pp. 3-29). Springer, Berlin, Heidelberg, 2008.

Christiansen, M., Decomposition of a combined inventory and time constrained ship routing problem. *Transportation science*, vol. 33, no. 1, pp.3-16, 1999.

Christiansen, M., Fagerholt, K., Nygreen, B. and Ronen, D., Maritime transportation. *Handbooks in operations research and management science*, vol. 14, pp.189-284, 2007.

Christiansen, M. and Fagerholt, K., Maritime inventory routing problems. In *Encyclopedia of optimization* (pp. 1947-1955). Springer, Boston, MA, 2008.

Christiansen, M., Fagerholt, K., Flatberg, T., Haugen, Ø., Kloster, O. and Lund, E.H., Maritime inventory routing with multiple products: A case study from the cement industry. *European Journal of Operational Research*, vol. 208, no. 1, pp.86-94, 2011.

Christiansen, M. and Nygreen, B., A method for solving ship routing problemswith inventory constraints. *Annals of operations research*, vol. 81, pp.357-378, 1998a.

Christiansen, M. and Nygreen, B., Modelling path flows for a combined ship routingand inventory management problem. *Annals of Operations Research*, vol. 82, pp.391-413, 1998b.

Cowling, P., Kendall, G. and Soubeiga, E.,. A hyperheuristic approach to scheduling a sales summit. In *International Conference on the Practice and Theory of Automated Timetabling* (pp. 176-190). Springer, Berlin, Heidelber, 2000.

Cowling, P., Kendall, G. and Soubeiga, E., Hyperheuristics: A tool for rapid prototyping in scheduling and optimisation. In *Workshops on Applications of Evolutionary Computation* (pp. 1-10). Springer, Berlin, Heidelberg, 2002.

Furman, K.C., Song, J.H., Kocis, G.R., McDonald, M.K. and Warrick, P.H., Feedstock routing in the ExxonMobil downstream sector. *Interfaces*, vol. 41, no. 2, pp.149-163, 2011.

Han, L., Kendall, G. and Cowling, P., An adaptive length chromosome hyper-heuristic genetic algorithm for a trainer scheduling problem. In *Recent Advances In Simulated Evolution And Learning* (pp. 506-525), 2004.

Hwang, S.J., *Inventory constrained maritime routing and scheduling for multi-commodity liquid bulk* (Doctoral dissertation, Georgia Institute of Technology), 2005.

Savelsbergh, M. and Song, J.H., Inventory routing with continuous moves. *Computers & operations research*, vol. 34, no. 6, pp.1744-1763, 2007.

Siswanto, N., Essam, D. and Sarker, R., Solving the ship inventory routing and scheduling problem with undedicated compartments. *Computers & Industrial Engineering*, vol. 61, no. 2, pp.289-299, 2011.

Siswanto, N., Wiratno, S.E., Rusdiansyah, A. and Sarker, R., 2018. Maritime inventory routing problem with multiple time windows. *Journal of Industrial & Management Optimization*, DOI: 10.3934/jimo.2018091, in press.

Song, J.H. and Furman, K.C., A maritime inventory routing problem: Practical approach. *Computers & Operations Research*, vol. 40, no. 3, pp.657-665, 2013.

Biography

Nurhadi Siswanto is a senior lecturer at the Industrial Engineering Department, Institut Teknologi Sepuluh Nopember, Indonesia. He has an industrial engineering bachelor degree from Institut Teknologi Sepuluh Nopember, Indonesia. He got a master degree in Industrial Engineering from Purdue University, USA and PhD from University of New South Wales, Canberra, Australia. His research interests include operation research and large scale optimization modeling, discrete event simulation modeling, and maritime transportation modeling.