ICIC Express Letters
Part B: Applications
Volume **11**, Number **10**, October **2020**

ICIC International ©2020 ISSN 2185-2766
pp. **987–994**

Q-LEARNING-BASED TECHNIQUE FOR REDUCTION OF NUMBER OF EMPTY-TRUCK TRIPS IN INTER-TERMINAL TRANSPORTATION

Taufik Nur Adi, Yelita Anggiane Iskandar and Hyerim Bae

Department of Industrial Engineering
Pusan National University
2, Busandaehak-ro 63beon-gil, Geumjeong-gu, Busan 46241, Korea
{ taufiknuradi; yelita_iskandar; hrbae }@pusan.ac.kr

Received March 2020; accepted June 2020

Abstract. *The vast growth in containerized trade has obliged major ports to add terminals for improved performance. The increasing number of terminals escalates the demand for movement of containers and cargo among facilities, which process is known as Inter-Terminal Transportation (ITT). A lack of adequate planning of ITT might lead to a major problem such as empty-truck trips which then increase ITT cost and decrease efficiency in the utilization of resources. To the best of our knowledge, the utilization of Reinforcement Learning (RL)-based approaches remains limited. Therefore, in this paper, we propose a Q-learning-based technique to reduce the number of empty-truck trips by considering a critical constraint such as the task time-window. Our result shows that the Q-learning-based technique finds 30.34% fewer empty-truck trips than simulated annealing and 94.42% faster in terms of computational time.*

Keywords: Inter-terminal transportation, Empty-truck trips, Q-learning

1. Introduction.

Containerized shipment plays an important role in global trade which has obliged most major ports to add terminals to satisfy the increasing container transport demand. At the same time, adding terminals escalates the demand for transportation of containers and cargo between terminals, which process is known as Inter-Terminal Transportation (ITT).

ITT is required between various origins and destinations within a seaport, which incorporates facilities such as warehouses, truck depots, repair stations, customs, screening and scanning facilities, and other dedicated transport terminals, as shown in Figure 1. Consequently, ITT forms a complex transportation network, which, if not handled efficiently, might become a major source of error, delay and inefficient resource utilization. [1,2] stated that the main objective of an efficient ITT system is to minimize or eliminate transport delay. Other objectives can be minimization of transport cost, handling time, or empty-truck trips, and maximization of occupancy rates. A review conducted by [4] showed that the empty-truck trip problem at container terminals is critical since trucks are the main mode of freight transportation there; indeed, truck's cargo space is limited, and when it remains empty, costs still incurred. Busan Port is the largest container port in South Korea, the fifth busiest in the world. In 1997, the South Korean government decided to construct a new port, named as Busan New Port (BNP) as in Figure 2, to alleviate cargo congestion.

BNP operates five container terminals with 23 berths. According to data collected in 2013, approximately 2600 containers were moved between the terminals each day [3]. In 2014, the Busan Port Authority (BPA), as an operator of Busan Port, initiated a project for surveying and analysis of the efficiency and quality of the ITT process. In

DOI: 10.24507/icicelb.11.10.987

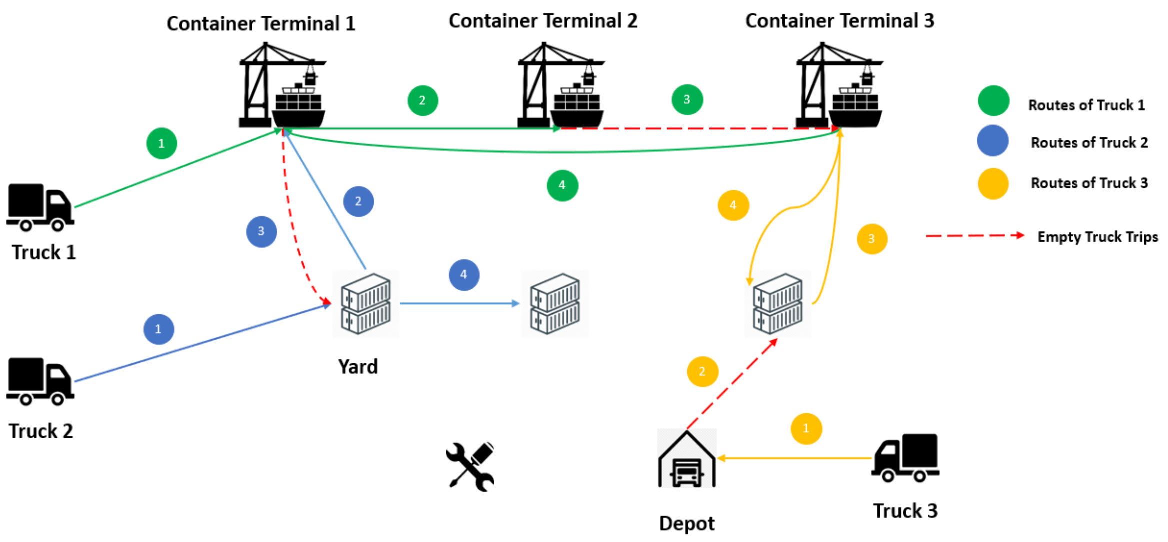

Figure 1. Inter-terminal transportation of containers

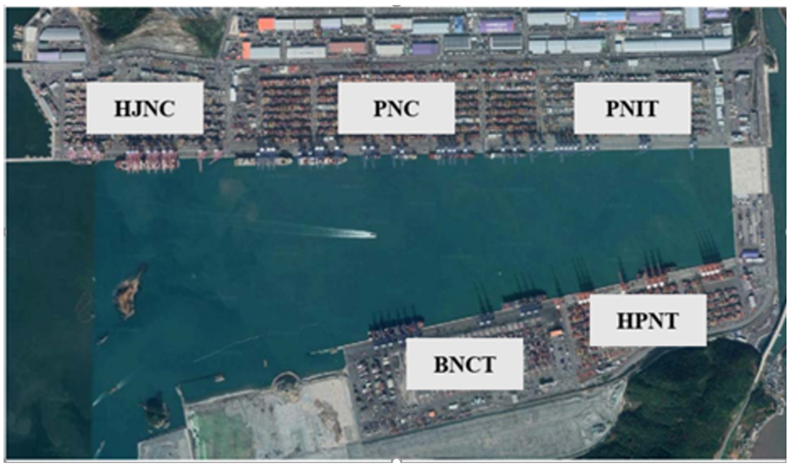

Figure 2. Layout of Busan New Port [3]

Busan area, ITT has suffered profit decrease during the last few years [5] indicating how critical an efficient ITT operation to the competitiveness of large seaport. Although many researchers have attempted to tackle the ITT-related problems, most of them used the optimization and metaheuristic approaches. To the best of our knowledge, the utilization of RL-based approaches remains limited. Therefore, this paper proposes Q-learning, a reinforcement learning algorithm, to minimize the number of empty-truck trips between container terminals.

This paper is organized as follows. Section 2 discusses the related work. Section 3 provides a problem description. Section 4 outlines the proposed Q-learning technique, and Section 5 presents the experimental results. Finally, Section 6 draws conclusions and future work.

2. Related Work.

In the literature, numbers of studies focusing on the reduction of the number of empty-truck trips can be found. [4] executed a comprehensive review of the empty-truck trip problem at container terminals. The author identified the causes, benefits and constraints, and proposed two collaborative approaches, namely Collaborative Logistics Network (CLN) and shared-transportation, to overcome the problem. [6] proposed a Mixed Integer Linear Programming (MILP) formulation to minimize trips overall costs subjected to time windows. They showed that trip combination can reduce costs as well as the number of empty-truck trips. [7] presented a model to reduce costs and number of empty-truck trips by taking advantage of truck capacity. They demonstrated that lack of coordination and collaboration between trucking companies in a port is one of the causes of empty-truck trips. [8] developed a Discrete-Event Simulation (DES) model to assess coordinated truck appointments and used it to solve the problem as a Traveling Salesman Problem (TSP) with time windows. They integrated the approach with a real-time Truck Appointment System (TAS) using data from Santo Antonio in Chile, to reduce polluting gas emissions. [9] simulated trucks sharing in a port environment. They found that collaboration among trucks increases port traffic but reduces gas emissions. RL has been widely used in solving problems related to scheduling and routing. [10-12] confirmed that RL is an efficient method to solve scheduling problems, while [13] used modified Q-learning to find more optimal travel plan for an on-demand bus system. Previous studies mentioned above have similar characteristics with ITT-related problem in terms of transportation mode, demand, capacity, order time-window, and trip plan. Nonetheless, RL has not yet been appropriately applied to ITT problem. The present study used Q-learning to design specific states, actions, and rewards to fit the objective of minimizing the total number of empty-truck trips at container terminals.

3. **Problem Description.**

Some essential information of ITT operations that should be provided includes demand, origin and destination terminals, delivery time windows, available modes and connections [14]. In our case, ITT demand, origin and destination terminals, delivery time windows, and available connections were known. For the transportation mode, we use only truck with one container capacity. Solution to the ITT problem here will satisfy the following restrictions: each transport task has to be performed, by only one truck, within given time window. At the beginning, the starting location of the truck is terminal 1. After the truck has served a task, its ending location will be the destination of that task. The objective is to produce a trip plan that has minimum empty-truck trips for a given tasks list.

4. **Proposed Q-Learning.**

Reinforcement learning is an unsupervised machine-learning technique dealing with a problem wherein autonomous agent can learn to select proper actions by interacting with its environment. Every time agent performs an action, the environment will return a reward or penalty. The agent will learn from these rewards and penalties in order to develop a decision-making policy. The Q-learning algorithm, proposed in [15], is one of the most utilized RL algorithms, with objective to learn a state-action pair value $Q(s,a)$ representing a long-term expected reward for each state-action pair.

$$Q(s_t,a) = (1-a)Q(s_t,a) + \alpha(r + \gamma^* \max[Q(s_{t+1},a)]) \tag{1}$$

where $Q(s_t,a)$ is the expected value of executing action a in state s; r is the immediate reward for executing action a; α is the learning rate; γ is the discount factor $(0 \leq \gamma \leq 1)$ that has an impact on the present value of future rewards; t is the stage of the action taken; $\max[Q(s_{t+1},a)]$ is the maximal value of Q under state s_{t+1}. In our case, the original Q-learning was inadequate because our state, action and reward depend on many characteristics, and not just on time-steps. Detailed definitions and explanations for each state, action, and reward in our design will be provided in the following sub-sections.

4.1. **Agent state.**

The state of the agent is defined as S_{pcn}, where p is the previous location of the agent, $p = \{1,2,3,4,5\}$. At the beginning, the value of $p = 0$ means that there is no available previous location. The symbol c is the current location of the agent. At first iteration, the value of $c = 1$. The last state element is n, the available task characteristics. The agent will encounter different available task characteristics each time state changes which is defined by comparing the current location of the agent to all available task origins. As shown in Figure 3(a), if current location of the agent is terminal 1 then compared to all available task origins, the agent faces two available task characteristics. The first characteristic is a task (task id 1) for which the task origin has the same location as current location of the agent. The second one is a task (task id 2 & 3) for which the task origin is not the same as its current location. Then, as shown in Figure 3(b), the agent is in terminal 3 and also encounters two available task characteristics, but the situation is different from the previous state. Task id 1 is in grey since it is already served; thus the remaining tasks are tasks 2 and 3, which respectively, have the opposite route contrasted to previous task, and have, as their task origin, one that is not the same as the current location of the agent.

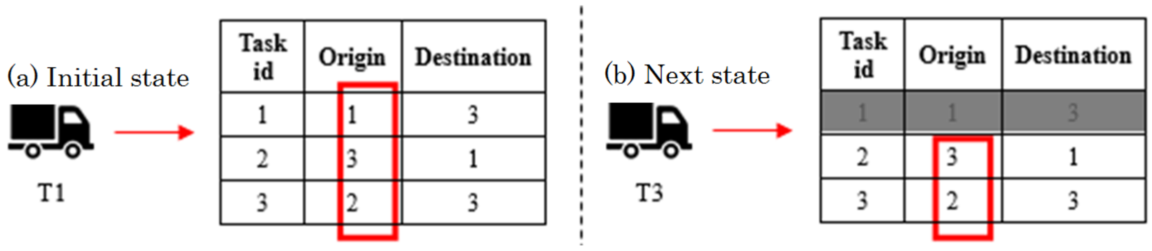

FIGURE 3. Available task characteristics

Table 1 shows the eight possible available task characteristics, which consists of all individual conditions and their combinations. Each available task characteristic is encoded as a number $(0, 1, \ldots, 7)$. State S013 means the agent has no previous location, its current location is terminal 1, and the available task characteristics have two conditions as stated in Table 1 No. 3. The state space of our Q-learning is 240 states, since p has six possible values $(0, 1, 2, 3, 4, 5)$, c has five possible values $(1, 2, 3, 4, 5)$, and n has eight possible values $(0, 1, \ldots, 7)$.

TABLE 1. Available task characteristics

No.	Available task characteristic combinations
0	Reverse route* Current agent location == order origin Current agent location != order origin
1	Reverse route* Current agent location == order origin
2	Reverse route* Current agent location != order origin
3	Current agent location == order origin Current agent location != order origin
4	Reverse route*
5	Current agent location == order origin
6	Current agent location != order origin
7	No available order

*previous task destination == next task origin and previous task origin == next task destination

4.2. **Agent actions.** A finite set of actions for an agent is defined as a_i for $i = 1, 2, 3, 4, 5$. a_1 means the agent is not taking any task or idle; a_2 means the agent takes a random task; a_3 means the agent takes a task that has reverse route with respect to previous task; a_4 means the agent takes a task for which its origin has the same location with current location; a_5 means the agent takes a task for which the task origin is not the same with current location. Considering the number of states and actions, Q-table used to store Q-values, forms a matrix with 240 rows and five columns.

4.3. **Rewards.** A reward in RL is a form of feedback from the environment. Through interaction with the environment, agent observes changes to the state and reward reflecting in selected action. The agent uses this reward to draw a conclusion on how to behave in certain state. In general, the objective is to learn what action should be taken at certain state to gain maximum reward. In our case, the objective is determining the task to be served to produce minimum number of empty-truck trips in ITT. Therefore, actions a_3 and a_4 will present the highest reward value producing maximum non-empty-truck trips. Table 2 indicates the complete reward values applied based on actions and available task characteristic criteria.

Table 2. Reward values with their criteria

Available task characteristics	Action	Reward value
$0, 1, 2, 4$	a_3	10
$0, 1, 3, 5$	a_4	
$0, 2, 3, 6$	a_5	1
$0, 1, 2, 3, 4, 5, 6$	a_2	
7	a_1	
$0, 1, 2, 3, 4, 5, 6$	a_1	-10

The reward value for choosing these two actions (a_3 and a_4) is based on the condition of the available task characteristics, 10. In other words, those actions are highly expected to be taken by the agent in any state that has available task characteristics $0, 1, 2, 4$ and $0, 1, 3, 5$, respectively. Additionally, at the initial stage, the agent will receive a reward value of -10 if he chooses action a_3.

Table 3. Example of task table

Task id	**Origin**	**Destination**	**Start time window**	**End time window**
1	1	3	0	15
2	3	1	0	60
3	2	5	20	80
4	...	...	...	...

5. **Experimental Results.** To assess the proposed method, we consider five scenarios with different numbers of tasks as presented in Table 4. We compare the result of the proposed algorithm with a metaheuristic algorithm, Simulated Annealing (SA), which uses Greedy Location Targeted Heuristic (GLTH) [16]. For Q-learning, the γ value is set to 0.9 and $\alpha = 0.01$ then run for 100 episodes. Gamma (γ) is a discount factor for future rewards, as a measure of how far ahead in time, the algorithm looks. To prioritize rewards in distant future, the value of γ should be set closer to one. The value of gamma indicates whether the agent prioritizes future or immediate reward. The alpha (α) refers to the learning rate. Larger value of alpha allows the model to learn faster while a lower one lets the model learn optimality more but may take longer. For SA, the initial temperature (α) is set to 100, the cooling rate (β) to 0.01, and the stopping criterion is set at $\alpha = 0$. The algorithm was implemented in Python and run on a PC equipped with an Intel(R) Xeon(R) CPU E3-1230 v5 of 3.40GHz and 16GB memory.

Based on Table 4, RL exhibits better performance in terms of computational time and finding the lowest number of empty-truck trips. The computational time of SA grows exponentially as the number of tasks increased. For the fifth scenario, with 180 tasks, RL was able to find 30.34% fewer trips than SA and was 94.42% faster in terms of computational time.

Table 4. Result comparison between reinforcement learning and simulated annealing

		Reinforcement learning		Simulated annealing	
No.	Tasks	Lowest number of empty-truck trips found	$t(s)$	Lowest number of empty-truck trips found	$t(s)$
1	10	4	0.7	4	1.6
2	20	10	1.43	9	6.2
3	40	19	6.64	21	50.23
4	80	29	9.22	34	125
5	180	62	59.46	89	1066

Figure 4 shows that cumulative reward increases dramatically between episode 0 and 10. It indicates that the agent learned faster in selecting task producing high reward.

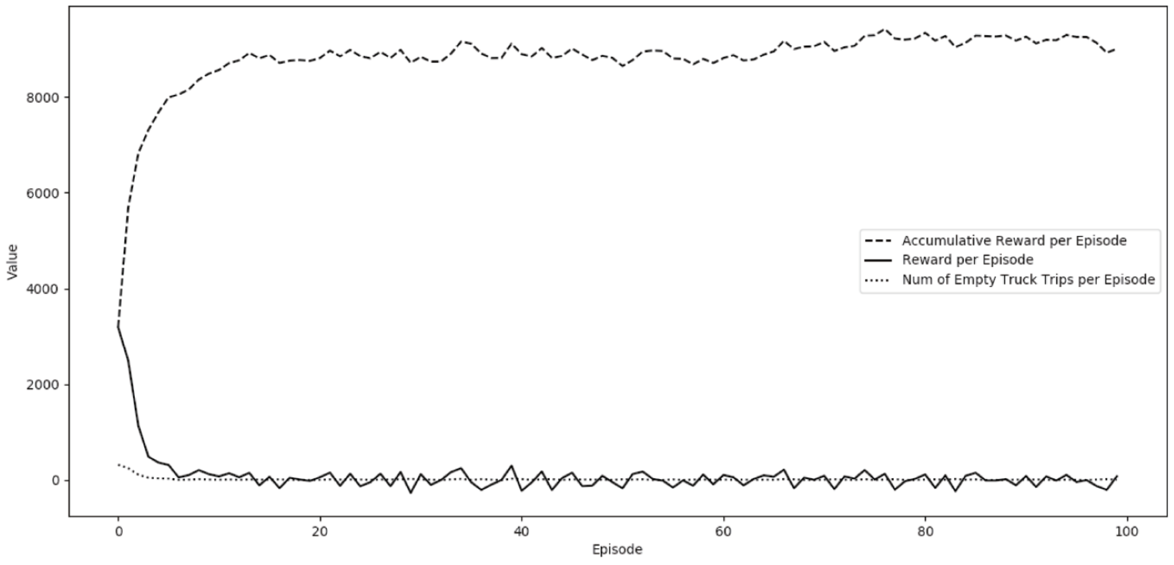

Figure 4. Reward, cumulative reward, and number of empty-truck trips per episode

Figure 5 shows that SA needs more time to find the lowest number of empty-truck trips. It could find the minimum number of empty-truck trips after the temperature falls below 37.

6. **Conclusions and Future Work.** Reinforcement learning could find the optimal solution quickly, because it learns from previous experience through reward system. RL has limitation when it learns only from a small number of experiences, and it tends to make random decisions on situations that have not been learned. The effectiveness of simulated annealing in finding optimal data relies on the effectiveness of the solution generation. Without it, SA tends to try all possible combinations that lead to inefficiency. The applicability of the proposed approach is still limiting with following assumptions: the number of container terminals is fixed (five terminals); the number of tasks is static and known in advance; and the trip plan only considers one available truck with one

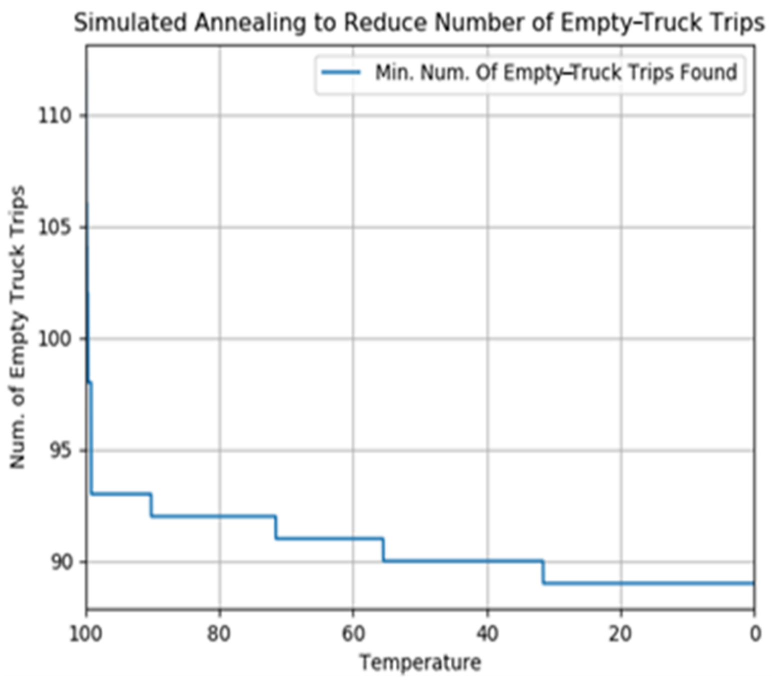

Figure 5. Change of number of empty-truck trips based on temperature

container load. More complex and realistic problems of ITT, such as multi-objective ITT routing, collaborative ITT, and dynamic trips planning will be solved in future research.

Acknowledgment. This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ICT Consilience Creative program (IITP-2019-2016-0-00318) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

REFERENCES

[1] M. Duinkerken, R. Dekker, S. Kurstjens, J. Ottjes and N. Dellaert, Comparing transportation systems for inter-terminal transport at the Maasvlakte container terminals, *OR Spectrum*, vol.28, no.4, pp.469-493, 2006.
[2] K. Tierney, S. Voß and R. Stahlbock, A mathematical model of inter-terminal transportation, *European Journal Operation Research*, vol.235, no.2, pp.448-460, 2014.
[3] X. Jin and K. Kim, Collaborative inter-terminal transportation of containers, *Industrial Engineering & Management Systems*, vol.17, no.3, pp.407-416, 2018.
[4] S. Islam, Empty truck trips problem at container terminals, *Business Process Management Journal*, vol.23, no.2, pp.248-274, 2017.
[5] H. Kopfer, D. Jang and B. Vornhusen, Scenarios for collaborative planning of inter-terminal transportation, *Lecture Notes in Computer Science*, pp.116-130, 2016.
[6] C. Caballini, I. Rebecchi and S. Sacone, Combining multiple trips in a port environment for empty movements minimization, *Transportation Research Procedia*, vol.10, pp.694-703, 2015.
[7] C. Caballini, M. Paolucci, S. Sacone and E. Ursavas, Towards the physical Internet paradigm: A model for transportation planning in complex road networks with empty return optimization, *Lecture Notes in Computer Science*, pp.452-467, 2017.
[8] F. Schulte, R. González and S. Voß, Reducing port-related truck emissions: Coordinated truck appointments to reduce empty truck trips, *Lecture Notes in Computer Science*, pp.495-509, 2015.
[9] S. Islam, Simulation of truck arrival process at a seaport: Evaluating truck-sharing benefits for empty trips reduction, *International Journal of Logistics Research and Applications*, pp.1-19, 2017.
[10] M. Aydin and E. Öztemel, Dynamic job-shop scheduling using reinforcement learning agents, *Robotics and Autonomous Systems*, vol.33, nos.2-3, pp.169-178, 2000.
[11] Y. Wang and J. Usher, Learning policies for single machine job dispatching, *Robotics and Computer-Integrated Manufacturing*, vol.20, no.6, pp.553-562, 2004.

[12] Q. Zeng, Z. Yang and X. Hu, A method integrating simulation and reinforcement learning for operation scheduling in container terminals, *Transport*, vol.26, no.4, pp.383-393, 2012.
[13] N. Mukai, T. Watanabe and J. Feng, Route optimization using Q-learning for on-demand bus systems, *Lecture Notes in Computer Science*, pp.567-574, 2008.
[14] L. Heilig and S. Voß, Inter-terminal transportation: An annotated bibliography and research agenda, *Flexible Services and Manufacturing Journal*, vol.29, no.1, pp.35-63, 2017.
[15] C. Watkins and P. Dayan, Q-learning, *Machine Learning*, vol.8, nos.3-4, pp.279-292, 1992.
[16] L. Heilig, E. Lalla-Ruiz and S. Voß, port-IO: An integrative mobile cloud platform for real-time inter-terminal truck routing optimization, *Flexible Services and Manufacturing Journal*, vol.29, nos.3-4, pp.504-534, 2017.