Scheduling multiple yard cranes in two adjacent container blocks with position-dependent processing times

Feng Chu, Junkai He, Feifeng Zheng, Ming Liu

Food safety risks in traditional fermented food from South-East Asia

Anil Kumar Anal[1,2], Giorgia Perpetuini[2,3,4], Awanwee Petchkongkaew[2,5], Reasmey Tan[2,6], Sylvie Avallone[7], Rosanna Tofalo[4], Hai Van Nguyen[2,8], Son Chu-Ky[2,8], Phu Ha Ho[2,8], Thanh Tam Phan[2,8], Yves Waché[2,3]

[1]Department of Food Agriculture and Bioresources, School of Environment, Resources and Development, Asian Institute of Technology, PO Box 4; Klong Luang 12120, Thailand

[2]Tropical Fermentation Network

[3]International Joint Laboratory « Tropical Bioresources & Biotechnology » (Univ. Bourgogne Franche-Comté, AgroSup Dijon, PAM UMR A 02.102 and School of Biotechnology and Food Technology, Hanoi University of Science and Technology), Dijon France

[4] Faculty of BioScience and Technology for Food, Agriculture and Environment, University of Teramo, Via R. Balzarini 1, 64100, Teramo, Italy

[5]School of Food Science and Technology, Faculty of Science and Technology, Thammasat University, 12120, Thailand

[6]Food Technology and Nutrition Research Unit, Research and Innovation Center, Institute of Technology of Cambodia, Russian Federation Blvd., P.O. Box 86, 12156 Phnom Penh, Cambodia

[7]QualiSud, Univ Montpellier, CIRAD, Montpellier SupAgro, Université d'Avignon, Université de la Réunion, Montpellier, France.

[8]International Joint Laboratory « Tropical Bioresources & Biotechnology » (Univ. Bourgogne Franche-Comté, AgroSup Dijon, PAM UMR A 02.102 and School of Biotechnology and Food Technology, Hanoi University of Science and Technology), Hanoi, Vietnam

Abstract

South-East Asia is well-known for traditionally fermented foods. However, these products are generally still produced at small scale following traditional procedures. Nowadays, consumers are particularly aware of the health concerns regarding food additives; the health benefits of "natural" and "traditional" foods, processed with no added chemical preservatives, are becoming more and more attractive. Therefore, their confidence towards safety and quality of Asian fermented foods is low. Major food safety concerns are related not only to food production methods, but also to how foods are processed, stored, sold and consumed. In this review the main factors affecting food safety are analysed. They are not limited only to the improper use of chemicals such as pesticides or antibiotics, but also to improper processing and handling during storage which could provoke the accumulation of toxic compounds such as mycotoxins or biogenic amines. Urgent attention is required to improve the quality of the ingredients and the integration of food safety management systems for industrial growth. Therefore, in the last part of this review directions to improve the food safety of fermented foods are proposed.

Keywords: fermented foods, South-East Asia, food safety

1. Introduction

In South-East Asia, the economic and demographic developments together with the migration of the population to urban areas resulted in many changes in the organisation of the food production system. This increased the consumer concern about food safety (Ha, Shakur, & Pham Do, 2019). However, the level of concern is not only coming from an undefined perception of this change but it follows also several outbreaks that occurred in the recent years. Drivers of food safety unconformity have been analysed (Kendall, Kaptan, Stewart, Grainger, Kuznesof, Naughton, et al., 2018). They have multiple origins among them the lack of resources, the economic pressure, the structuration, the training and the will. As confirmed by the achievement of the Asifood Erasmus+ project (Anal, Waché, Louzier, Roy, Mens, Avalllone et al. submitted), the demand of actors of the field for food safety is huge. Among the various food products concerned, the case of traditional fermented food is special as these foods are usually produced at small scale, very popular and considered as delicacies in Asia but also, often as risky (Sarter, Ho, & To, 2014). Taking into account the classification proposed by Steinkraus (Steinkraus, 1997), popular fermented food in continental South-East Asia (Valyasevi & Rolle, 2002; Vu & Nguyen, 2016) are belonging to the (i) High salt/meat-flavored amino acid/peptide sauce (various fish sauces or pastes like nuoc mam, prahok, pla ra, soy sauces like Tuong), (ii) Lactic acid fermentations (vegetable leaves, bamboo, onions, fermented meat that can be uncooked (Nem chua, nahm), fish, shrimp), (iii) alcoholic fermentation (rice wine), (iv) acetic fermentation (rice vinegar), (v) alkaline fermentation of soy (thua-nao). With the general economic idea of decreasing the risks and increasing the scale of production, these products could be the victim of a food diversity extinction. However, other economic models exist like in Europe where small-scale fermented foods are considered as a way to increase typicity and bring value-added.

The goal of this review is to evaluate the dangers and risks to which traditional fermented food are exposed in South-East Asia. Recent studies, from 2007 to 2019, concerning the detection of pathogens and chemical contaminants in some of the most risky products are reported. From these results, directions to improve the food safety of fermented foods in Vietnam, Cambodia and Thailand will be proposed.

2. Contamination of raw or fermented products with pathogenic bacteria

In South-East Asia, and in particular in Vietnam, consumers' confidence towards fermented products is low, especially because of the microbial safety risks of the raw material such as meat and vegetable (Sarter, Ho, & To, 2014).

One of the riskiest traditional fermented meat products is an uncooked pork sausage eaten after a short lactic fermentation. It is called *nem chua* in Vietnam, *nahm* in Thailand and *nem chrouk* in Cambodia. To elaborate this product, pork is bought in the wet market or supermarket and then the minced meat is mixed with herbs and spices such as garlic, guava leaf, fresh chili, etc. The finish products are packed into banana leaves to provide an anaerobic environment for the fermentation process and to inhibit entry of potentially pathogenic microorganisms. The products are stored at room temperature for spontaneous fermentation for several days.

The highest risk of *nem chua* is related to meat since this fermented sausage is uncooked and unheated. In pork raw meat from South-East Asia, *Escherichia coli*, *Salmonella, Campylobacter, Listeria monocytogenes* and *Staphylococcus aureus* are the main food-borne pathogens and they are frequently found in intestinal tract and faeces of food animals (Dao & Yen, 2006) (Nguyen Thi Nhung, Van, Cuong, Duong, Nhat, Hang, et al., 2018) (Ananchaipattana, Hosotani, Kawasaki, Pongsawat, Md.Latiful, Isobe, et al., 2012; Carrique-Mas, Bryant, Cuong, Hoang, Campbell, Hoang, et al., 2014; Dang-Xuan, Nguyen-Viet, Unger, Pham-Duc, Grace, Tran-Thi, et al., 2017; T. N. M. Nguyen, Hotzel, El-Adawy, Tran, Le, Tomaso, et al., 2016; Takeshi, Itoh, Hosono, Kono, Tin, Vinh, et al., 2009; Toan, Nguyen-Viet, & Huong, 2013). These bacteria as well as antibiotic resistance genes can be horizontally transferred to human through direct and indirect contacts with the source of infection which can be animals or contaminated foods. In raw food, in the examples given in Table 1, these bacteria are found with at least 10% probability of contamination. The probability could rise up to 83% depending on the species, the raw food, and the origin of the sample.

In fermented meat, the contamination is still very high in the examples presented in Table 1 and they even reach 100% in the neighboring regions of India (Keisam, Tuikhar, Ahmed, & Jeyaram, 2019) although in one study not presenting whether fermented meat was cooked or not, contaminations ranged from 0 to 19% (Ananchaipattana, et al., 2012). A study focusing on *Salmonella* in the food chain and in nem chua confirmed through serovar analysis and genotyping that contamination of the raw meat was usually responsible for contamination of the fermented product (Le Bas, Hanh, Thành, Cuong, Quang, Binh, et al., 2008). The microbial safety of *nem chua* in Vietnam has been discussed in several studies ((Phan, Pham et al. 2006; Nguyen et al., 2010; 2013 ; Le, Do et al. 2012). Indeed, the concentration of *E. coli* and *S. aureus* in raw meat for preparing *nem chua* was 10-100 and 100-1000 fold higher than the National Vietnamese Standard (TCVN 7046:2002 for fresh meat), respectively. Consequently, the final products *nem chua* could not meet the requirement for hygiene and safety standard (TCVN 7050:2002 for unheated fermented products) (Le, Do, Le, Tran, & Van, 2012; Phan, Pham, & Hoang, 2006). However, cases of illness in human beings due to the consumption of *nem chua* have beeen rarely reported in Vietnam. Several explanations can contribute to this fact, including small scale production combined with incomplete epidemiological data, as well as the presence of a number of lactic acid bacteria (LAB) present in the final product. These LAB and their metabolites can produce organic acid and bacteriocins able to inhibit the growth of pathogenic microorganisms. The LAB of 30 samples of *nem chua* were isolated and the presence of *Lactobacillus plantarum* (prevalence of 67.6%), *Pediococcus pentosaceus* (21.6%), *Lactobacillus brevis* (9.5%) and *Lactobacillus farciminis* (1.3%) was reported (Tran, May, Smooker, Van, & Coloe, 2011). *L. plantarum* was also reported as the most prevalent in meat and legume fermentation products (La Anh, 2015; D. T. L. Nguyen, Van Hoorde, Cnockaert, De Brandt, Aerts, & Vandamme, 2013). Some authors investigated thus the antibacterial properties of some *L. plantarum* strains isolated from fermented food. *L. plantarum* B33 isolated from *nem chua* could inhibit food-borne bacteria including *E. coli* NC31, *E. coli* K12TG1, *E. coli* 320 LCB, *S. aureus, S.* Typhimurium with a zone of inhibition ranging from 4-14 mm (Lê, Hồ, Trần, Chu, Lê, Lê, et al., 2011). Recently, the bacteriocins produced by other LAB isolated from *nem chua* were characterized (Pilasombut, Rumjuankiat, Ngamyeesoon, & Duy, 2015). It was shown that *L. plantarum* KL-1 could produce a bacteriocin inhibiting the growth of pathogens and of some LAB such as *Lactobacillus sakei, Leuconostoc mesenteroides* and *Enterococcus faecalis*. Additionally, 47% LAB isolates from *nem chua* exhibited strong antimicrobial activity against moulds from the same products and *L. plantarum* and *P. pentosaceus* showed the best antifungal activities (Phong, Van, Thanh, Long, & Dung, 2016). These strains could thus be useful as backslop and/or starter cultures.

Vegetables can also be the source of microbiological contamination (Ha et al., 2019). In South-East Asia, there are different kinds of fermented vegetables including cucumbers, mustard greens, young melons, cabbages, chinese cabbages, papayas, bamboo shoots, and bean sprouts with Asian spider flower. They are usually produced in small-scale using mostly spontaneous fermentation and sometime back-slopping fermentation. The production of fermented foods and beverages through spontaneous fermentation and back-slopping represents an inexpensive and reliable preservation method in less developed countries (Owens, 2014). Most of the sellers of these fermented vegetables are producers and these products are usually sold in the local markets in open containers. The results of two studies carried out in Phnom Penh wet markets are reported in Table 1.

In these studies, the microbiological quality of fermented vegetables sold in Phnom Penh markets was investigated, showing a correlation between microbial contamination and hygienic conditions (quality and material of container, use of hands, hands with gloves, the same rice paddle for all the products, open containers, etc…). The authors supposed that the lack of hygienic precautions and bad cultural practices could be extended upward to the production of vegetables (González García, Fernández-López, Polesel, & Trapp, 2019).

This paragraph deals with pathogen bacteria but it should be noted that another source of disease has been characterized in Thai fermentd fish. Indeed, liver fluke in its development cycle is ingested by fishes that, if eaten without cooking, can inoculate the disease in human consumers (Sriraj, Boonmars, Aukkanimart, Songsri, Sripan, Ratanasuwan, et al., 2016). It is particularly dangerous as liver fluke can be a factor of development of cholangiocarcinoma, especially when it is linked to high concentrations of nitrosamine (Mitacek, Brunnemann, Suttajit, Martin, Limsila, Ohshima, et al., 1999). However, it can be noted that the level of such cancers is not higher in North-East Thailand, the region where this fermented fish is produced and consumed.

3. Toxin or toxic compounds producing microorganisms

3.1 Mycotoxin risks

Mycotoxins are toxic secondary fungal metabolites mainly produced by five genera of filamentous fungi i.e. *Alternaria, Aspergillus*, *Cladosporium, Fusarium* and *Penicillium*. These molds can produce different types of mycotoxins, among them, some are unique to one species, but most can be produced by several fungi (Bräse et al., 2009). FAO reported that 25% of the world's crops are affected by mold or fungal growth with losses of around 1 billion metric tons of food products annually. Economic losses occur because of: 1) yield loss due to diseases induced by toxigenic fungi; 2) reduced crop value resulting from mycotoxin contamination; 3) losses in animal productivity from mycotoxin-related health problems; and 4) human health costs. Moreover, additional costs associated with mycotoxin also include the cost of management at all levels (prevention, sampling, mitigation, litigation, and research costs). These economic impacts affect all along the food and feed supply chains: crop producers, animal producers, grain handlers and distributors, processors, consumers, and the society as a whole (due to health care impacts and productivity losses). Nowadays, more than 400 mycotoxin metabolites have been discovered but the eight most important mycotoxins (Top 8) with worldwide relevance in regard to public health are aflatoxins, ochratoxin A, fumonisin B_1, zearalenone, deoxynivalenol, nivalenol, T-2 toxin and patulin (FAO, 2001). Mycotoxin can contaminate various agricultural commodities and particularly cereals and legumes such as wheat, barley, rye, oats, rice, maize, peanuts, alfalfa, clover, beans, peas, chickpeas, lentils, lupine, mesquite, carob, soybeans, tamarind and some other cereal grains that are normally used as substrates for traditional fermented food in South-East Asia. Those substrates can be contaminated with mycotoxin either before harvest or under post-harvest conditions (FAO, 1991), resulting in mycotoxin contamination in the finished products. Mycotoxin is relatively stable to cooking and processing temperatures. Once they contaminate food and feed, they cannot be removed safely. This means that once foods are contaminated, human exposure is almost certain if the foods go into the market.

In South-East Asia, this problem is acute due to favorable humidity and temperature conditions for the development of molds. It is even hypothesized that the consumption of aflatoxin contaminated foods, which is recognized as a risk factor for human hepatocellular carcinoma, may contribute to the high incidence of this disease in South-East Asia (Tran-Dinh, Kennedy, Bui, & Carter, 2009). Several studies report a high level of contamination of agricultural products and, due to the great stability of mycotoxins, these compounds remain present in agriculture soils (Tran-Dinh, Kennedy, Bui, & Carter, 2009). They also remain in the raw product after processing and, if the raw material is used for animal feeding, they can contaminate meat products. As a result, despite a 20-year-old study showing in the north of Thailand that vegetarians can be more exposed to mycotoxins (Vinitketkumnuen, Chewonarin, Kongtawelert, Lertjanyarak, Peerakhom, & Wild, 1997), animal products are also likely to be contaminated. For instance in Vietnam, aflatoxins and zearalenone were found in all feed samples analyzed (Thieu, Ogle, & Pettersson, 2007) and then aflatoxin M1, in more than half of pig urine samples collected in various slaughterhouses of Vietnam (Lee, Lindahl, Nguyen-Viet, Khong, Nghia, Xuan, et al., 2017). As a result, mycotoxins can be expected in fermented products in a way similar to unfermented ones (Sivamaruthi, Kesika, & Chaiyasut, 2018). Although analyses of mycotoxins present in fermented products are rare, some have been detected in soy fermented products like Thua-nao (Petchkongkaew, Taillandier, Gasaluck, & Lebrihi, 2008) and Tuong (Vu & Nguyen, 2016). This latter soy sauce is fermented by *Aspergillus orizae* which is very near, and considerd as the domesticated form of, *A. flavus*, the aflatoxin producer, a fact that can explain the presence of aflatoxin in 4/14 brands of Tuong. A review presenting mycotoxins in world fermented products has been published recently confirming that mycotoxins are often present in this class of products (Sivamaruthi, Kesika, & Chaiyasut, 2018).

3.2 Biogenic amines

Biogenic amines (BAs) are low molecular weight nitrogenous compounds naturally present in animals, plants, and microorganisms where they play several functions including gene expression regulation, cell growth and differentiation, etc. (Suzzi and Torriani, 2015). On the basis of their chemical structure, BAs are divided into aliphatic (putrescine, cadaverine, spermine and spermidine), aromatic (tyramine and phenylethylamine) or heterocyclic (histamine and tryptamine). Considering the number of amine groups, they are classified in monoamines (tyramine and phenylethylamine), diamines (putrescine and cadaverine) or polyamines (spermine and spermidine) (Park et al., 2019). The occurrence of some BAs (histamine, putrescine, cadaverine, tyramine, tryptamine, 2-phenylethylamine, spermine and spermidine) in fermented foods such as fish, meat, cheese, vegetables, and wines, has been widely described (for a review see Spano et al., 2010). Unfortunately, the consumption of foods or beverages containing high amounts of BAs is a risk for consumer health since they can have toxic effects (Park et al., 2019). The BAs encountered in fermented foods are mainly produced by microbial decarboxylation of amino acids (Mah et al., 2019). The main microbial groups associated with BAs accumulation are several Gram negative (enterobacteria and pseudomonads) and Gram positive bacteria including staphylococci, *Bacillus* spp. and lactic acid bacteria (LAB). The presence of decarboxylase positive microorganisms is not the only factor influencing BAs accumulation in foods, in fact, specific environmental conditions are required (e.g. availability of BAs precursors, presence of proteolytic enzymes involved in the release of free amino acids). Other factors involved are: raw materials characteristics in terms of composition, pH, ion strength, physico-chemical parameters (NaCl, pH and ripening temperature) and processing, storage and distribution conditions (Suzzi and Torriani, 2015; Linares et al., 2012).

Several traditional fermented foods from South-East Asia including fish sauce and fermented soybean foods are characterized by a high amount of BAs (for reviews see Prester, 2011, Zaman et al., 2009; Park et al., 2019). The most popular fermented soybean foods are produced through bacterial fermentation and the main are: Natto, Miso, Cheonggukjang, Doenjang, Gochujang, Chunjang, Doubanjiang, and Douchi. The BAs composition of these products is reported in Table 2 (it should be noted that for Doenjang, the very high maximal concentration detected comes from one isolated study and most studies report values below thresholds).

Fermented fish products represent a source of BAs and especially histamine, which is one of the most dangerous for human health. In fact, it is the only BA with regulatory limits. The US FDA set a guidance level of 50 mg/kg for histamine in the edible portion of fish (FDA, 2011), and the European Commission established up to a maximum of 200 mg/kg in fresh fish and 400 mg/kg in fishery products treated by enzyme maturation in brine (EFSA, 2011). BA content in seafood for South-Korea and China is regulated imposing maximum limits of histamine content in fish, at 200 mg/kg and 200–400 mg/kg, respectively. According to EFSA (2011) dried anchovies and fish sauce are the fermented foods showing the highest mean content of histamine, with values of 348 mg/kg and 196-197 mg/kg, respectively. More in general, fish sauce is characterized by the highest mean values for the sum of BAs (582 - 588 mg/kg). The histamine content of some Asian fish products is reported in Table 3.

4. Chemical contaminations

4.1 Pesticides, heavy metals

In the past century, agriculture has increased productivity through mechanization, fertilization, pesticides, and selective breeding. Furthermore, intensification of cash-crop production and conventional agriculture with fertilizers and pesticides impair local resources (soil fertility, biodiversity). Some trace elements (e.g. iron, potassium, etc…) have undergone historical decrease in food in Finland, United States and United Kingdom (Mayer, 1997; Ekholm et al., 2007). This decline was attributed to varietal selection based mainly on yield and soil degradation due to intensive agriculture.

Great transitions are at work in Europe. The Ecophyto plan is implemented with the purpose of progressively reducing the use of pesticides. With international trades, food products are imported from numerous countries where the environmental rules do not correspond to European ones. In Northern Europe, foods imported from Asia contained 111 distinct pesticides and in some cases concentration exceeded the Maximum Residue Levels with leafy vegetables particularly concerned (chives, thai basil) (Skretteberg, 2015). In 2010, among 245 plant foods from Phnom Penh markets, 15% of the long beans and 95% of the kale contained noticeable levels of organophosphate and carbamate pesticides (Neufeld et al., 2010).

It is true that, apart from the war inheritances like agent orange which still results in daily intakes of dioxins far above the WHO recommendation in some parts of Vietnam (Tuyet-Hanh, Minh, Vu-Anh, Dunne, Toms, Tenkate, et al., 2015), South-East Asian cultures can be potentially contaminated by chemicals and metals. Several studies have highlighted the poisoning of agriculture workers (Thetkathuek & Jaidee, 2017; Thetkathuek, Suybros, Daniell, Meepradit, & Jaidee, 2014) and of the environment (Harnpicharnchai, Chaiear, & Charerntanyarak, 2014) but the main concern for fermented food products is related to pesticides encountered in raw materials. A survey in the Red river delta showed that pesticides used in agriculture were frequently detected in biota, leading to repeated analyses above the acceptable daily intakes in fishes and vegetables (Hoai, Sebesvari, Minh, Viet, &

Renaud, 2011). Even banned organochlorine pesticides still persists in these environments. Among vegetables reaching the residue limits established by the European Union, some vegetables used for fermentation like chinese cabbage and some herbs are cited (Sapbamrer & Hongsibsong, 2014; Wanwimolruk, Kanchanamayoon, Phopin, & Prachayasittikul, 2015) while some fruits (watermelon and durian) despite the presence of chemicals, were considered as safe (Wanwimolruk, Kanchanamayoon, Boonpangrak, & Prachayasittikul, 2015). The fate of pesticides in food products depends on the specific physicochemical properties of the compounds as well as of the conditions of preparation of food (presence of water, temperature and pH etc). During fermentation, the concentration of pesticides usually decreases significantly, giving rise to degradation products. In addition to non fermented products, fermented products are characterized by the presence of microorganisms. The presence of pesticides as well as of their degradation products tend to limit the activity of microorganisms and modify the sensorial properties of the product but in the mean time, microorganisms can degrade these products and help decreasing the contamination of fermented products (Regueiro, López-Fernández, Rial-Otero, Cancho-Grande, & Simal-Gándara, 2015). Studies on pesticides in food are still at the preliminary level and their degradation, the risk related to their degradation products, the potential catalysis by microorganisms, thought of paramount importance, need to be further studied.

5. Antibiotics resistance of microbes from traditional fermented foods from South-East Asia

Antibiotics are either microbial secondary metabolites or the analogous compounds synthesized or semi-synthesized chemically, that could inhibit the growth and survival of other bacteria. These compounds are used as therapeutic agents against infectious disease in humans, livestock and aquaculture. However, haphazard and extensive use of antibiotics may select antibiotic resistant bacteria (Ben et al., 2019). Bacteria can develop antibiotic resistance by several mechanisms via enzymatic degradation; antibiotic target modification; changing the bacterial cell wall permeability and alternative pathways to escape the activity (Verraes et al., 2013). Antibiotic resistance can be inherited or acquired. Inherited antibiotic resistance is exhibited by all isolated of the same species while acquired antibiotic resistance occurs when susceptible bacteria gain the genes encoding a resistance mechanism via mutation or the transfer of genetic material from other bacteria (MacGowan & Macnaughton 2017).

Foodborne diseases not only affect people's well-being, but also cause hospitalization and economic loss. Approximately 22.8 million cases of diarrheal illness caused by Salmonellosis outbreak annually, with 37,600 deaths in South-East Asia (Van et al., 2012). Generally, food contributes as an important part for transfer of antibiotic resistance in terms of antibiotic residues or resistant genes from food microflora to pathogenic bacteria (Akbar & Anal, 2013). It is important to monitor the prevalence of pathogenic bacteria along with antibiotic resistant foodborne pathogens in food chain to improve and implement food safety (Chanseyha et al., 2018). For instance, for the contamination of raw products by pathogenic bacteria as discussed above, the resistance of bacteria to antibiotic is thus also an important issue. The resistance of *E. coli* to antibiotic isolated from pork collected in pig farm, supermarket and wet market from Hochiminh city and Tien Giang province was reported. The resistance to tetracycline, sulphafurazole, ampicillin/amoxicillin, trimethoprim, chloramphenicol, streptomycin, nalidixic acid, ciprofloxacin, gentamicin, colistin and ceftazidime were 100, 70, 55, 60, 50, 65, 30, 42.2, 73.3, 22.2 and 1.1%, respectively (N. T. Nguyen, Nguyen, Nguyen, Nguyen, Nguyen, Thai, et al., 2016; Van, Chin, Chapman, Tran, & Coloe, 2008). Nguyen et al. (2016a) described for the first time a strain – isolated from pork – resistant against colistin (a last-resort antibiotic). Multi-resistance of *E. coli* to at least 3 different classes of antibiotics was observed with rates up to 75% in pork (Van, Chin, Chapman, Tran, & Coloe, 2008).

Campylobacter and *Salmonella* spp. from pork meat showed high resistance rate (>50%) against streptomycin and tetracycline in addition, with different levels of antimicrobial susceptibility to ciprofloxacin, nalidixic acid, ampicillin, chloramphenicol, erythromycin and streptomycin (0-<50%) (T. N. M. Nguyen, et al., 2016; Vo, van Duijkeren, Gaastra, & Fluit, 2010). Moreover, 52.2% of *Salmonella* strains from pork, beef and chicken meat showed multidrug resistance (Nguyen Thi Nhung, et al., 2018).

Traditional fermented foods and beverages are popular for their nutritional balance and food security. In many Asian countries, techniques for fermenting cereals, vegetables and meat products are well developed. Such fermented foods are highly priced for improved nutritional and organoleptic quality as well as for beneficial microorganisms (Anal, 2019). Several studies have indicated that along with beneficial microorganism, fermented foods can act as vehicles of antibiotic resistant bacteria (see Table 4). Therefore, through the food chain involving traditional fermented food, antibiotic resistant genes may be transferred to other bacteria including pathogens and commensals and into the gastrointestinal tract (Abriouel et al., 2017).

South-East Asia region is rapidly developing, and food products are exported globally. Since this region is a hotspot of antibiotic resistant bacteria, there is a risk of dissemination of antibiotic resistant bacteria and genes to consumers worldwide (Nhung et al., 2016). In Asia, numerous fermented foods are categorized into five groups including fermented soybean; fish; vegetable; bread and porridges; and alcoholic beverage. Lactic acid bacteria (LAB) are the commonly involved bacteria in these products to a varying extent, having either positive or negative effects (Rhee *et al.,* 2011). Some of the antibiotic resistant LAB isolated from traditionally fermented foods are summarized in the Table 4.

6. Analysis and main recommendations

Despite huge differences between South-East Asian countries, concerns about risk of pathogens in fermented meat product such as *nem chua,* due to the contamination in raw meat material, are common (Hoang & Vu, 2017). The high level of contaminations in farms can be related to hygiene (Dang-Xuan, Nguyen-Viet, Pham-Duc, Unger, Tran-Thi, Grace, et al., 2019). However, several studies have highlighted an increase in the level of contaminations in slaughterhouses. These contaminations were likely to come from faeces to carcasses (Dang-Xuan, et al., 2019; Le Bas, et al., 2008). Cross contaminations between species have also been observed as shown in Table 1: although infections by *Campylobacter* concerned mostly poultry and poultry products, this species was also found in swine carcasses and retail products as the meat could be contaminated with faeces at the slaughterhouse and processing facilities during the evisceration process, leading to the contamination of food products (T. N. M. Nguyen, et al., 2016). Taking into account that pigs are still mainly slaughtered in small structures in Vietnam despite the existence of big modern structures, one axis of improvement could be the improvement of hygiene conditions in those small slaughterhouses in a way sustainable for small structures. These recommendations for good hygiene practices could be: (1) separate gut rinsing and carcass dressing, (2) separate lairage and carcass dressing, (3) use specific working surface for the carcass during slaughter, (4) cleaning and disinfection of these surfaces after work, (5) give clean water to the pigs at lairage (well water, not tank water), (6) clean and disinfect tanks and tools after work, (7) waste management avoiding contamination (Le Bas, et al., 2008). Other steps can be causes of contaminations such as handling in pork shops where management of flies and handling avoiding contact between meat and worker clothes could be improved (Dang-Xuan, et al., 2019). Finally, although the original load of raw pork meat is certainly the main contributor to the pathogen contamination in the sausage, some additional contaminations may also result from the processing, especially during the forming of the sausages into the guava and banana leaves, usually done by hand. Leaves being usually considered as the source of inoculation of fermenting lactic acid bacteria, a maintained spontaneous fermentation would require an improvement of the leave production while another perspective would be the use of starters. Recommendation could also target consumers and a Thai study recommended also to cook nahm before eating (Paukatong & Kunawasen, 2001), however, such a step would introduce a completely new way of consumption in Vietnam.

For vegetables, basic hygiene could improve significantly the situation concerning pathogens. Contrary to the general consumer perception, hygienic problems are not limited to wet markets but are also present in supermarkets. For instance, a Thai study detected almost no significant contamination differences between open markets and supermarkets however, it could not take into account differences concerning fermented products as they were only available in open markets at that time (Ananchaipattana, et al., 2012). It is likely that a training of producers and handlers to good practices could improve greatly the hygienic quality of fermented products. One particular care should be taken to cross contaminations which have also been observed in vegetables as mixed vegetables are far more contaminated than fermented monoproducts. Moreover, the use of hurdle technologies like biopreservation, involving the use of efficient starters could also contribute to improve pathogenic control through sustainable solutions (Ho, 2017).

The problem of mycotoxin comes mainly from conditions of mold development, which can be decreased through the respect of good practice, and from the high stability of concerned molecules that can stay in the environment a long time after the production by molds. One concern is the price of analyses, which makes difficult any actions for basic farmers. There are however also some solutions to decrease the level of contamination by using adsorbant materials like bentionite clays for piglet farming (Thieu, Ogle, & Pettersson, 2008) or microbial catalysts able to degrade the toxins (Petchkongkaew, Taillandier, Gasaluck, & Lebrihi, 2008).

The fluctuations in the BAs concentration suggest the lack of standardized processes. Improvements in this direction could be useful. In this context, it is essential to implement control strategies and develop methods to reduce BAs content in fermented products in order to face consumers demand for healthier and safe foods.

Food contamination by pesticides is a world-wide problem but, concerning specific contaminations in the region, the agricultural practices are obviously a way to improve results as, for mangosteen, 97% of the fruits for local market are exceeding the MRLs whereas no problems are detected for the GAP production of fruits intended for the European Union (Wanwimolruk, Kanchanamayoon, Phopin, & Prachayasittikul, 2015). Organic agriculture could also be a mean to improve this point but some steps are still required to secure production especially in less controlled area. Indeed, the safety of organic foods is still unclear as the use of untreated irrigation or washing water or inappropriate organic fertilizers (i.e manure or composts) is possible (Taban et al., 2011; Nguyen-the et al, 2016). Data related to the microbial communities associated to the surface of fresh fruits and vegetables have shown significant differences, quantitatively and qualitatively, between conventional and organic products (Leff & Fierer, 2013; Bigot et al., 2015). Using *E. coli*, *Salmonella* and *Listeria* which are the most contaminating pathogen bacteria for leafy vegetables as indicator, different studies have been carried out to evaluate the contamination level of organic vs conventional vegetables, notably leafy ones (Tango et al., 2014; Karp et al., 2016). Results are contrasted with no real trend or a slight impact on organically-grown vegetables (Leff & Fierer, 2013; Tango et al., 2014). Studies on the mycotoxin contamination of agricultural products have mainly been carried out on cereals or fruits and rarely on leafy vegetables (van der Walt et al., 2006; Sanzani et al., 2016). Mycotoxin contaminations in organic cereal crops were variable and inconclusive but their levels were sometimes higher than those observed in conventional samples (Baert et al., 2006; Magkos et al., 2006; Piqué et al., 2013). In this field, more studies are needed before establishing precise recommendations.

In all cases, training of farmers, retailers and transformators would greatly improve the situation.

7. Conclusion

Fermented foods from South-East Asia exhibit several safety risks in a similar way to raw agriculture products (Figure 1). The presence of pathogens is one of the main and immediate risk as it contaminates many clean samples during slaughter and handling. However, the presence of chemicals, biogenic amins, mycotoxins and parasites can be a more difficult risk to take into account as it can be more hidden with a long-term effect. Risks could be greatly diminished by the use of good agriculture/manufacturing practice, especially concerning pathogen and chemical contamination and antibiotic resistance. For mycotoxin, good producing and storage practice could also help a lot and, for biogenic amines, a survey of strains present during fermentation might be required as a first step before strategies to avoid their development. For all food safety concerns, starters can also be used for biopreservation and bioremediation purposes and thus decrease risks (Ho, Nguyen, Petchkongkaew, Nguyen, Chu-Ky, Nguyen, et al., in minor revision).

This validates the drivers of food unsafety as cited above (Kendall, et al., 2018). Among the various ways to improve safety, training of all actors of the production/transformation chain may be one of the main active ones. This was the subject of the AsiFood Erasmus+ project which just finished recently (Anal et al., in press).and, in the future, it would be interesting to evaluate the impact of training on food safety in the different partner countries. It should also be added that the level of food safety in the various ASEAN countries is very diverse as a result of economic development, culture, motivation (export to world region exerting a big control) etc.

Acknowledgements : This work was funded by the Erasmus+ AsiFood «Universities as key partners for the new challenges regarding food safety and quality in ASEAN» project, the Bualuang ASEAN chair professorship of Thammasat University, the Investissement d'Avenir ISITE BFC ANR-15-IDEX-0003 and the French Embassy in Thailand. The author are thankful to Dr Guillaume Da for his precious help to establish the network.

References

Abriouel, H., Knapp, C. W., Gálvez, A., & Benomar, N. (2017). Antibiotic resistance profile of microbes from traditional fermented foods. In J. Frias, C. Martinez-Villaluenga, & E. Peñas (Eds.), Fermented Foods in Health and Disease Prevention (pp. 675–704). Academic Press.

Akbar, A., & Anal, A. K. (2013). Prevalence and antibiogram study of *Salmonella* and *Staphylococcus aureus* in poultry meat. *Asian Pacific journal of tropical biomedicine*, 3, 163–168.

Ananchaipattana, C., Hosotani, Y., Kawasaki, S., Pongsawat, S., Md.Latiful, B., Isobe, S., & Inatsu, Y. (2012). Prevalence of Foodborne Pathogens in Retailed Foods in Thailand. *Foodborne Pathogens and Disease*, 9(9), 835-840.

Anal, A. K. (2019). Quality Ingredients and Safety Concerns for Traditional Fermented Foods and Beverages from Asia: A Review. *Fermentation*, 5, 8.

Anal, A. K., Waché Y., Louzier V., Roy L., Mens, F. Avalllone, S., Mahakarnchanakul W, Udompijitkul P., Tantikitti C, Nguyen, T. B. T., Phan T.P., Nguyen, T, M, T, Nguyen H.M.X., Kong T, Hul, S., Schleining G, Lindner L, Schippo, M.L., Guidi, A (in press). AsiFood and its Output and Prospects: An Erasmus+ Project on Capacity Building in Food Safety and Quality for South-East Asia. *Food Control*.

Baert, K., De Meulenaer, B., Kamala, A., Kasase, C., & Devlieghere, F. (2006). Occurrence of patulin in organic, conventional, and handcrafted apple juices marketed in Belgium. *Journal of Food Protection,* 69, 1371–1378.

Ben, Y., Fu, C., Hu, M., Liu, L., Wong, M. H., & Zheng, C. (2018). Human health risk assessment of antibiotic resistance associated with antibiotic residues in the environment: A review. *Environmental Research*, 169, 483–493.

Bigot, C., Meile, J.C., Kapitan, A., & Montet, D. (2015). Discriminating organic and conventional foods by analysis of their microbial ecology: An application on fruits. *Food Control*, 48, 123–129.

Binh, D. X., Minh, N. N., & Nguyet, D. T. (2017). Prevalence of Listeria monocytogenes, E. coli, Salmonella spp. and Staphylococcus aureus Bacteria Contamination on Meat at Public Market in the North of Vietnam. *SOJ Microbiology & Infectious Diseases*, 5, 1–22.

Bräse, S., Encinas, A., Keck, J., & Nising, C. F. (2009). Chemistry and Biology of Mycotoxins and Related Fungal Metabolites. *Chemical Reviews*, 109, 3903–3990.

Carrique-Mas, J. J., Bryant, J. E., Cuong, N. V., Hoang, N. V., Campbell, J., Hoang, N. V., Dung, T. T., Duy, D. T., Hoa, N. T., Thompson, C., Hien, V. V., Phat, V. V., Farrar, J., & Baker, S. (2014). An epidemiological investigation of Campylobacter in pig and poultry farms in the Mekong delta of Vietnam. *Epidemiology and Infection*, 142, 1425–1436.

Chanseyha, C., Sadiq, M. B., Cho, T. Z. A., & Anal, A. K. (2018). Prevalence and analysis of antibiotic resistant genes in *Escherichia coli* and *Salmonella* isolates from green leaf lettuce. *Chiang Mai Journal of Science*, 45, 1274–1286.

Chrun, R., Hosotani, Y., Kawasaki, S, & Inatsu, Y. (2017). Microbioligical Hazard Contamination in Fermented Vegetables Sold in Local Markets in Cambodia. *Biocontrol Science*, 22, 181–185.

Dang-Xuan, S., Nguyen-Viet, H., Pham-Duc, P., Unger, F., Tran-Thi, N., Grace, D., & Makita, K. (2019). Risk factors associated with *Salmonella* spp. prevalence along smallholder pig value chains in Vietnam. *International Journal of Food Microbiology*, 290, 105-115.

Dang-Xuan, S., Nguyen-Viet, H., Unger, F., Pham-Duc, P., Grace, D., Tran-Thi, N., Barot, M., Pham-Thi, N., & Makita, K. (2017). Quantitative risk assessment of human salmonellosis in the smallholder pig value chains in urban of Vietnam. *International journal of public healt*h 62, 93–102.

Dao, H. T. A. & Yen, P. T. (2006). Study of Salmonella, Campylobacter, and Escherichia coli Contamination in Raw Food Available in Factories, Schools, and Hospital Canteens in Hanoi, Vietnam. *Annals of the New York Academy of Sciences*, 1081, 262–265.

Ekholm, P., Reinivuo, H., Mattila, P., Pakkala, H., Koponen, J., Happonen, A., Hellstrom, J., & Ovaskainen, M. L. (2007). Changes in the mineral and trace element contents of cereals, fruits and vegetables in Finland. *Journal of Food Composition and Analysis*, 20, 487–495.

Ezzat, M. A., Zare, D., Karim, R., & Ghazali, H. M. (2015). Trans- and cis-urocanic acid, biogenic amine and amino acid contents in ikan pekasam (fermented fish) produced from Javanese carp (Puntius gonionotus) and black tilapia (Oreochromis mossambicus). *Food Chemistry,* 172, (2015) 893–899.

González García, M., Fernández-López C., Polesel F., & Trapp S. (2019). "Predicting the uptake of emerging organic contaminants in vegetables irrigated with treated wastewater – Implications for food safety assessment." Environmental Research 172, 175–181.

Ha, T. M., Shakur, S., & Do, K. H. P. (2019). Consumer concern about food safety in Hanoi,Vietnam. *Food Control,* 98, 238–244.

Harnpicharnchai, K., Chaiear, N., & Charerntanyarak, L. (2014). Residues of organophosphate pesticides used in vegetable cultivation in ambient air, surface water and soil in Bueng Niam Subdistrict, Khon Kaen, Thailand. *The Southeast Asian journal of tropical medicine and public health*, 44, 1088–1097.

Scheduling multiple yard cranes in two adjacent container blocks with position-dependent processing times

Abstract

This paper studies the management of three yard cranes in two adjacent container blocks in line, where cranes can move from one block to the other. Comparing with existing literature, the new multi-yard-crane scheduling problem incorporates different constraints together: (i) three yard cranes are deployed simultaneously in two adjacent blocks, (ii) non-crossing and inter-crane interference constraints of yard cranes are considered, (iii) the processing time of each container depends on its real-time location, i.e., position-dependent processing times. For the problem, a 0-1 mixed integer programming (MIP) model is constructed to minimize the total flow time to reduce the total container storage time in container yards, which helps to save container yard resources and increase production efficiency. This proposed model can be solved optimally by CPLEX for small-size instances. As the concerned problem is NP-hard, a fast heuristic and an improved genetic algorithm are devised to produce near-optimal solutions for large-size instances. Numerical experiments validate the developed MIP model and demonstrate the efficiency of the proposed algorithms.

Keywords: Yard crane scheduling; Interference; Position-dependent processing times; Mixed integer programming; Heuristics

1. Introduction

Maritime transportation is growing rapidly due to its larger load capacity and lower price compared with other transportation modes. As a result, maritime transportation demand and seaborne trade volume continue to increase with global economic and merchandise trade growth (c.f., UNCTAD 2016). For example, world container port throughput increases to 601.8 million 20-foot equivalent units (TEUs) in 2012. Recent statistics show that container trade volumes reach 160 million TEUs with an average growth of 4.6% per year (Liu et al., 2016). Container terminals act different roles for container activities, such as processing import, export and transshipment tasks. Under the context of heavy workload and limited resources, many scholars seek effective methods to schedule equipment so as to balance the demand and the supply (Chen et al., 2013).

Typically, containers are temporarily stored in dedicated container blocks. In a container block, yard rubber-tired gantry cranes (yard cranes for short) are the main equipment to handle containers. The study shows that yard crane productivity usually cannot meet the retrieval demands due to the unreasonable deployment (c.f., UNCTAD 2016). Besides, Li et al. (2009) point out that container terminal operations are often bottlenecked by slow yard crane movements that increase the operation time of yard cranes, as well as the waiting time of trucks and vessels. Consequently, how to effectively deploy yard cranes for processing containers (tasks) is of great importance for daily operations in container terminals.

In order to enhance the working efficiency, two yard cranes are employed in a single container block for processing containers (Ng, 2005). However, according to the practical applications, there is not always possible to deploy two yard cranes in a single yard block due to a lack of key resources (i.e., yard cranes). For example in Shanghai Port, there are around 70 yard cranes but more than 100 yard blocks. On the other hand, exporting containers take more than 60% of all container tasks, which undoubtedly brings great challenges for limited yard-crane resources. To our best knowledge, at export container zones, the scheduling of three yard cranes in two adjacent blocks is becoming common in practice. Meanwhile, this scheduling problem has not been studied in previous literature which is more complex than our previous work with two yard cranes in a single block (Zheng et al, 2018a).

The following phenomenon cannot be ignored in daily operations. The first one is the consideration of inter-crane interference. Because yard cranes are huge and bulky, any two adjacent yard cranes have to keep a given distance during processing to avoid mutual influence and guarantee the safety. Thus, the interference of any two cranes should be respected. The second phenomenon is that non-target containers need to be removed (i.e., non-productive reshuffling operations) such that target containers can be retrieved. As a result, the processing time of a target container is variable according to its dynamic position in a stack. However, most existing works ignore this feature and assume a fixed container processing time. In this paper, we focus on variable processing times and denote it as *position-dependent processing times*.

Based on the above observation, this paper investigates a three-yard-crane scheduling problem within two adjacent blocks. Our contributions mainly include: (i) a new multiple-yard-crane scheduling problem is studied with position-dependent processing times and inter-crane interference, (ii) a 0-1 mixed integer programming (MIP) model is constructed for the proposed problem, (iii) problem properties are analyzed and efficient solution methods are devised for effectively solving the problem.

The remainder of this paper is organized as follows. Section 2 gives a brief literature review. In Section 3, the considered problem is described in detail. A 0-1 mixed integer programming model is proposed in Section 4, and two heuristics are devised for solving the problem in Section 5. In Section 6, numerical experiments are conducted to evaluate the performance of the proposed methods. Section 7 concludes this paper and indicates future research directions.

2. Literature review

The management of container terminals attracts more and more attention in recent years. The research interests contain the seaside optimization, container yard optimization, and integrated optimization (Bierwirth and Meisel, 2010; Vis et al., 2010; Zhen et al., 2011; Carlo et al., 2014; Liu et al., 2015a; Liu et al., 2015b; Zhen, 2016; Zheng et al., 2018b). As we focus on the scheduling of three yard cranes in two blocks, therefore, we first review the multiple-yard-crane scheduling problems in container terminals, and then production scheduling problems with position-dependent processing times.

2.1. Multiple-yard-crane scheduling problems

Certainly, the quay-crane related optimization problems in the seasides can provide some ideas for the yard-crane scheduling problems (Liu et al., 2016, Zheng et al., 2018a). However, the yard-crane scheduling in container yards has its specific characteristics. In general, the yard-crane scheduling can be distinguished by two types, i.e., the scheduling of a single yard crane and multiple yard cranes. The single-yard-crane scheduling problem is usually easier than the multiple ones since it mainly studies the loading or unloading operations in a container block (Ng and Mak 2005; Li et al., 2009; Huang and Guo, 2013; Huang et al., 2015).

Ng (2005) first studies a multiple-yard-crane scheduling problem in which two yard cranes in a container block are admitted. Besides, inter-crane interference is considered in his work. For the problem, an integer programming (IP) model is established and a lower bound is proposed. After that, many authors investigate multiple-yard-crane problems in a single container block to enhance the operational efficiency in a container yard. To be specific, Cao et al. (2010) consider a two-hybrid flow shop scheduling with two cranes and yard trucks in a container block. For this problem, an IP model is formulated and a Benders-cut based method is devised. Jian et al. (2014) study a multiple-yard-crane deployment problem in storage yards. The objective to minimize yard crane operating cost and inter-block movement cost. An IP model is formulated and a divide-and-conquer based heuristic is employed to solve this problem. Wu et al. (2015) investigate a two-yard-crane scheduling problem within a yard block and consider the operational restrictions, such as the crane non-crossing constraint and the crane travel time. A clustering-reassigning based heuristic is proposed for solving this problem. Gharchgozli et al. (2015) study a multiple-yard-crane scheduling problem. The problem is equivalent to a multiple asymmetric generalized traveling salesman problem. A MIP model with the makespan minimization is constructed and an adaptive neighborhood search heuristic is applied to quickly obtain near-optimal solutions.

The above works ignore the fact that a container processing time can be variable due to its real-time position. To the best of our knowledge, Zheng et al. (2018a) is the only work that considers variable container processing times. However, their work focuses on the scheduling of two yard cranes in a single container block. Nevertheless, two or more container blocks in line sharing yard cranes is a practice in container terminals. Moreover, the corresponding scheduling problem has not been studied.

2.2. Production scheduling with position-dependent processing times

Many scholars have considered production scheduling with position-dependent processing times. Guinet (2007) addresses an identical parallel-machine scheduling with position-dependent jobs, where a machine setup time, related to the job sequence, is required between job processing. The objective is to minimize the maximum completion time of the jobs. A heuristic based on Hungarian method is proposed for solving the problem. Zhu et al. (2011) study several single-machine scheduling problems where job processing times depend on their position. The problems for minimizing the weighted sum of makespan and total resource cost are proved to be polynomially solvable. Dolgui et al. (2012) consider a single-machine scheduling problem with precedence constraints in which the job processing time depends on its position. The objective is to minimize the makespan. It is proved that the problem can be solved in polynomial time. Hsu and Yang (2014) analyze serval unrelated parallel-machine scheduling problems with deteriorating effects and position-dependent processing times. The objectives include the weights of the total load, total completion time, total deviation of completion time, and total resource cost. They show that these problems are polynomial time solvable when the number of machines is fixed. Yin et al. (2014) assume that the job processing time depends not only on the starting time of the job but also on its scheduled position. The objective of the single-machine scheduling problem is to minimize the makespan. Several properties that can guarantee the optimality of this problem are provided. Furthermore, Yin et al. (2017) study a parallel-machine scheduling with position-dependent processing times, meanwhile, the deteriorating effect of jobs is also considered. The problem complexity is proved and a polynomial-time algorithm and approximation schemes are proposed.

The above works with position-dependent processing times focus on production scheduling problems, and crane scheduling characteristics have been considered. Consequently, the developed methods cannot be applied directly to crane scheduling problems in container terminals.

Summing up, considering that it is unlikely to always assign two cranes to a single container block during heavy workload periods, but two neighbor blocks in line usually can share three yard cranes at a time. In this paper, the twin-yard-crane scheduling in one block is extended to a three-yard-crane scheduling for two blocks in line, according to practical applications. The operation security feature (inter-crane interference) and real-time location of container (position-dependent processing times) are considered in this work.

3. Problem statement

In this section, the main characteristics of retrieval operations in two adjacent container yards are presented in detail.

Assume that the target containers and yard cranes can be considered as tasks and parallel machines, respectively (Xu and Yang, 2013). During the planning horizon T (consists of several equal-length time steps, indexed by t), there are $|N|$ containers for loading or retrieval by $|K|$ YCs in two adjacent container blocks. It is expected to determine the assignment of yard cranes and the processing sequence of containers to minimize the total flow time, i.e., the total holding time of all containers, to enhance the productivity in container blocks.

The layout of containers directly influences the decision making of yard-crane assignment (Do et al., 2014; Zhang et al., 2014; Zhen et al., 2016), Therefore, we first introduce the container location and crane operating environment in our work. In Figure 1, two adjacent blocks are labeled as 1 and 2. Their slot sets for them are denoted as $[b_1^L, b_1^R]$ and $[b_2^L, b_2^R]$, respectively. For example, $b_1^L = 1, b_1^R = 40$ and $b_2^L = 47, b_2^R = 86$ in Shanghai port. Yard crane 1 (YC1) and yard cane 3 (YC3) can only process tasks in blocks 1 and 2, respectively. However, yard crane 2 (YC2) can move between the two adjacent blocks. Therefore, YCs 1 and 2 can simultaneously process tasks in block 1, and so do YCs 2 and 3 in block 2. Two practical conditions are considered in this work: (i) Non-crossover constraint. YCs 2 and 3 (resp. YCs 1 and 2) cannot handle any task to the left (resp. right) of YC1 (resp. YC3) at any time. (ii) Inter-crane interference. There exists a distance of e between two blocks, and a safety distance d between any two cranes. For example in Figure 1, $e = 6$ slots and $d \geq 8$ slots.

Typically, the containers are temporarily stored in a container block. For example, each block contains 40 slots, 6 rows, and 6 layers according to the layout of Shanghai

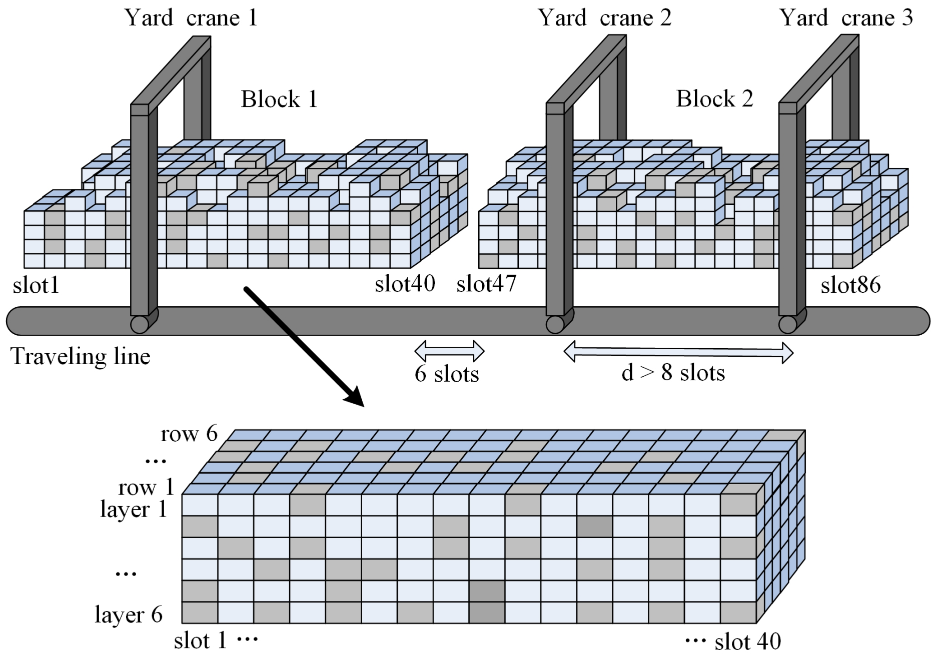

Fig. 1 Three yard cranes in two adjacent blocks.

Port. Figure 1 also gives a brief illustration of a block in which each shaded cube represents a task to be handled. Three-dimensional coordinates, including the row, slot, and layer, are used to describe the location of a container. In this way, the location of target container i is represented by a three-tuple (a_i, b_i, h_i), indicating respectively the row, slot, and layer of container i. In this work, we explicitly consider the dynamic container layer changing caused by reshuffling operations and position-dependent processing times. That is, the processing time of a container depends on its real-time vertical layer. Without loss of generality, Figure 2 explains how to calculate the processing time of a container.

In Figure 2, the top layer is labeled with the smallest index. The target containers j, m and i to be processed in stack 1 locate at layer 2, 3 and 5, respectively. In order to calculate their processing time (occupied time steps), we use u to represent the sum of time steps for a single container move. For example, container j needs $|3u|$ time steps to its processing because it consists of three single container moves, including removing the non-target container at layer 1, picking up container j and moving back the non-target container. Therefore, the processing time of task j is equal to $p_j = u \cdot (2h_j - 1)$, where h_j is the layer of container j. After processing containers j and m, the layer of container i changes from layer 5 in stack 1 to layer 3 in stack $1'$. Consequently, the

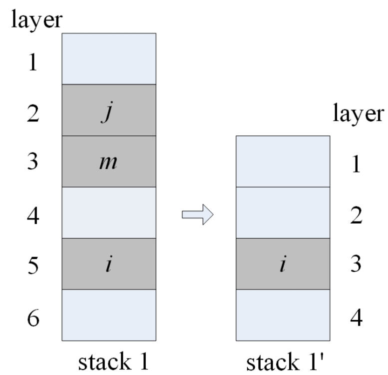

Fig. 2 The real-time renewal of layer indices of containers.

processing time of task i is $|5u|$.

This study is based on the following fundamental assumptions.

(i) Each task can be only assigned to one yard crane, each yard crane can handle one task at a time.

(ii) Interruption and preemption are always forbidden during processing.

(iii) In the beginning, YC1 is at the left end of block 1, YCs 2 and 3 are at the left and right end of block 2, respectively.

(iv) The traveling time of any YC between two adjacent slots can be computed according to the slot's length.

(v) All containers above each unloading target container must be moved back to the original stack immediately after the target container has been handled.

(vi) During the total planning horizon, the operation of these yard cranes can be not continuous. That is, idle time may exist in the schedule of YCs.

The problem consists of assigning containers to yard cranes and determining the scheduling of each crane. The objective is to minimize the total flow time of all containers, i.e., the sum of the deviation between the completion time and release time of each container. The release time of a target container represents the earliest time when it can be processed.

4. Mathematical model

In this section, a 0-1 mixed integer programming model is constructed for the considered problem. Before presenting the MIP model, the definitions of problem parameters and decision variables are first presented as follows.

4.1. Notation

Indices

- i, j: indices of tasks, i.e., target containers;

- k: index of three yard cranes, and $k = 1, 2, 3$ from left to right;

- b: index of slots;

- t: index of discrete time steps;

- z: index of container blocks in line.

Input parameters

- N: set of tasks;

- K: set of yard cranes;

- B: set of slots;

- T: set of time steps in the planning time horizon;

- Z: set of container blocks;

- M: a sufficiently large number;

- a_i: the row location of task i;

- b_i: the slot location of task i;

- $[b_z^L, b_z^R]$: the slot set of block z;

- h_i: the initial layer location of task i. If i is on the top of the stack, then $h_i = 1$;

- r_i: the release time of task i;

- u: the constant processing time of a single container operation (or movement) by a yard crane;

- d: the minimum safety distance between two adjacent YCs;
- $t_{i,j}$: the traveling time of YC from slot b_i of task i to slot b_j of task j, where $i,\ j \in N$, $i \neq j$;
- $\beta_{i,j}$: equaling to 1 if $a_i = a_j$, $b_i = b_j$ and $h_i < h_j$, where $i,\ j \in N$, $i \neq j$. That is, tasks i and j are at the same stack and task i is located above task j, 0 otherwise;
- $\gamma_{i,j}$: equaling to 1 if the slot location of task i is to the left of task j, where $i,\ j \in N$, $i \neq j$, i.e., $b_i < b_j$, 0 otherwise.

Decision Variables

- s_i: the start time of task i;
- c_i: the completion time of task i;
- $X_{i,k}$: equaling to 1 if task i is assigned to YCk, 0 otherwise;
- $Y_{i,j,k}$: equaling to 1 if tasks i and j are assigned to YCk, and task i is processed before j, where $i,\ j \in N$, $i \neq j$, $k \in K$, 0 otherwise;
- $O_{i,k,t}$: equaling to 1 if YCk processes task i in time step t, 0 otherwise;
- $W_{k,b,t}$: equaling to 1 if YCk occupies slot b in time step t, 0 otherwise;
- $\delta_{k,z,t}$: equaling to 1 if YCk processes in block z in time step t, 0 otherwise;
- $\varphi_{i,t}$: equaling to 1 if task i is started for processing in time step t, 0 otherwise;
- f_i: the number of tasks which are located above task i in the same stack and have been handled before task i.

4.2. Model formulation

To be specific, section 4.2.1 describes the objective function of the MIP model. Sections $4.2.2 - 4.2.4$ present the constraints of this problem. Ranges of decision variables are specified in Section 4.2.5. Besides, we use "$\{i \neq j\} \in N$" to concisely represent tasks i and j belong to set N while they indicate different tasks.

4.2.1. Objective function

The objective is to minimize total flow time, i.e., the sum of the completion time minus the release time of each task.

$$\min \sum_{i \in N} (c_i - r_i). \tag{1}$$

4.2.2. Task allocation constraints

$$\sum_{k \in K} X_{i,k} = 1, \qquad i \in N. \tag{2}$$

$$Y_{i,j,k} + Y_{j,i,k} \leq 1, \quad \{i \neq j\} \in N, k \in K. \tag{3}$$

$$Y_{i,j,k} + Y_{j,i,k} \geq -1 + X_{i,k} + X_{j,k}, \quad \{i \neq j\} \in N, k \in K. \tag{4}$$

$$X_{i,k} \geq Y_{i,j,k}, \quad \{i \neq j\} \in N, k \in K. \tag{5}$$

Constraint (2) ensures that each task i can be served by one exact YC. Constraint (3) describes that if tasks i and j are assigned to the same YCk, they must be processed in sequence, i.e., task i is processed before j or j is processed before i. The subsequent constraints (4) and (5) links the two binary parameters, meaning that if tasks i and j have a sequential relationship on YCk, they must have been assigned to YCk.

4.2.3. Time-related allocation constraints

$$\sum_{i \in N} O_{i,k,t} \leq 1, \qquad k \in K, t \in T. \tag{6}$$

$$O_{i,k,t} \leq X_{i,k}, \quad i \in N, k \in K, t \in T. \tag{7}$$

$$\phi_{i,t} \geq \varphi_{i,t}, \qquad i \in N, t \in T. \tag{8}$$

$$\sum_{t'=1}^{t-1} \phi_{i,t'} \leq M(1 - \varphi_{i,t}), \qquad i \in N, t \in T. \tag{9}$$

$$\sum_{t \in T} \varphi_{i,t} = 1, \qquad i \in N. \tag{10}$$

$$O_{i,k,t} \geq X_{i,k} + \varphi_{i,t} - 1, \quad i \in N, k \in K, t \in T. \tag{11}$$

$$\sum_{k \in K} O_{i,k,t} = \phi_{i,t}, \qquad i \in N, t \in T. \tag{12}$$

$$O_{i,k,t} - O_{i,k,t+1} \geq -M \cdot (2 - \phi_{i,t} - \phi_{i,t+1}), \quad i \in N, k \in K, t \in T. \tag{13}$$

$$O_{i,k,t} - O_{i,k,t+1} \leq M \cdot (2 - \phi_{i,t} - \phi_{i,t+1}), \quad i \in N, k \in K, t \in T. \tag{14}$$

$$k + 1 \leq k' + M \cdot (3 - O_{i,k,t} - O_{j,k',t} - \gamma_{ij}), \quad \{i \neq j\} \in N, \{k \neq k'\} \in K, t \in T. \tag{15}$$

$$\phi_{i,t+1} + \phi_{i,t-1} - \phi_{i,t} \leq 1, \quad i \in N, t \in T. \tag{16}$$

$$\sum_{b \in B} W_{k,b,t} = 1, \quad k \in K, t \in T. \tag{17}$$

$$W_{k,b_i,t} \leq 1 + M \cdot (1 - O_{i,k,t}), \quad k \in K, i \in N, t \in T. \tag{18}$$

$$W_{k,b_i,t} \geq 1 - M \cdot (1 - O_{i,k,t}), \quad k \in K, i \in N, t \in T. \tag{19}$$

Constraint (6) guarantees that each YC can process at most one task in each time step. Constraint (7) means that if YCk processes task i in time step t, then i must be assigned to this YC. Constraints (8) and (9) ensure that if task i is started for processing in time step t, then it is being processed in the following time steps rather than any of previous time steps. Constraint (10) represents that each task i can only be started once at one exact time step. Constraint (11) states that if task i has been assigned to YCk and started for processing in time step t, then $O_{i,k,t} = 1$. Constraint (12) describes the relationship between the two variables. Constraints (13) and (14) state that if task i is being processed in several consecutive time steps, it must be handled by the same YC. In constraint (15), if YCs k and k' are processing tasks i and j respectively in time step t, at the same time, task i is on the left slot of j, then YCk must be on the left side of k' during processing. Constraint (16) ensures that each task is processed in successive time steps. It means if task i is under processing in time steps $t - 1$ and $t + 1$, it must occupy t. Constraint (17) states that each YC occupies exactly one slot in any time step t. Constraints (18) and (19) assure that if YCk is processing task i in time step t, then the YC must locate at the slot of task i.

4.2.4. Retrieval processing time

$$s_i \geq r_i, \quad i \in N. \tag{20}$$

$$s_i = \sum_{t \in T} t \cdot \varphi_{i,t}, \quad i \in N. \tag{21}$$

$$f_j = \sum_{i \in N \setminus j} Y_{i,j,k} \cdot \beta_{i,j}, \quad j \in N. \tag{22}$$

$$u \cdot [2(h_i - f_i) - 1] = \sum_{t \in T} \phi_{i,t}, \quad i \in N. \tag{23}$$

$$c_i = s_i + \sum_{t \in T} \phi_{i,t} - 1, \quad i \in N. \tag{24}$$

$$s_j \geq c_i + t_{i,j} + 1 - M \cdot (1 - Y_{i,j,k}), \quad \{i \neq j\} \in N, k \in K. \tag{25}$$

Constraint (20) means that each task starts no earlier than its release time. Constraint (21) determines the start time step of task i. In constraint (22), variable f_j counts the number of target containers located above container j in stack (a_j, b_j), and have been handled before task j. Constraint (23) calculates the processing time of task i using variable f_i. Specifically, the processing time for a container can be calculated according to the description in Section 3, which is also equal to its occupied time steps. Constraint (24) computes the completion time of task i. Here we need to explain why it needs to minus 1. For example, if task i is started in time step 4, its processing time is 2, then task i is being processed in time steps 4 and 5. Therefore, the completion time is 4+2-1=5. In constraint (25), the start time of any task should be not earlier than the completion time of its previous task plus the traveling time of YC, provided that they are assigned to the same YC.

4.2.5. Crane position and interference constraints

$$b_i \leq b_1^R + M \cdot (1 - X_{i,1}), \qquad i \in N. \tag{26}$$

$$b_i \geq b_2^L - M \cdot (1 - X_{i,3}), \qquad i \in N. \tag{27}$$

$$\sum_{b=b_1^L}^{b_1^R} W_{2,b,t} + \delta_{2,2,t} = 1, \qquad t \in T. \tag{28}$$

$$\sum_{b=b_2^L}^{b_2^R} W_{2,b,t} + \delta_{2,1,t} = 1, \qquad t \in T. \tag{29}$$

$$b' \sum_{b' \in B} W_{2,b',t} - b \sum_{b \in B \setminus b'} W_{1,b,t} \geq d - M \cdot (1 - \delta_{2,1,t}), \quad i \in N, t \in T. \tag{30}$$

$$b' \sum_{b' \in B} W_{3,b',t} - b \sum_{b \in B \setminus b'} W_{2,b,t} \geq d - M \cdot (1 - \delta_{2,2,t}), \quad i \in N, t \in T. \tag{31}$$

With respect to Assumption (iii), constraints (26) and (27) regulate that YC1 and YC3 can only process tasks in blocks 1 and 2, respectively. Nevertheless, YC2 can handle tasks in both two adjacent blocks. In constraints (28) and (29), it is seen that if YC2 is processing task i (in slot b) at block 1 in time step t, then it cannot be at block 2 (i.e., $\delta_{2,2,t} = 0$), and vice versa. The safety distance constraint is guaranteed by constraints (30) and (31): if YCs 1 and 2 process tasks in slots b and b' respectively in block 1, within time step t, then the distance between them is at least d $(= 8)$, which is similar for YCs 2 and 3 in block 2.

4.2.6. Variable ranges

The range of decision variables are as follows:

$$s_i, c_i, f_i \in Z^+, \qquad \forall i \in N. \tag{32}$$

$$X_{i,k}, Y_{i,j,k}, \in \{0,1\}, \qquad \forall \{i \neq j\} \in N, k \in K. \tag{33}$$

$$O_{i,k,t}, \varphi_{i,t}, \phi_{i,t} \in \{0,1\}, \qquad \forall i \in N, k \in K, t \in T. \tag{34}$$

$$W_{k,b,t}, \delta_{k,z,t} \in \{0,1\}, \quad \forall k \in K, b \in B, t \in T, z \in Z. \tag{35}$$

To the best of our knowledge, the scheduling of a single yard crane with different task ready times has been proved to be NP-complete (Ng, 2005). Our newly addressed scheduling problem is more complicated since we further consider three yard cranes within two container blocks and position-dependent processing times. Without a doubt, our studied problem also has NP-hard nature.

For exactly solving the above MIP model, the commercial solver, CPLEX, is applied. The experimental environment and computational results are specified in Section 6. It can be observed that when the input size increases a little, from 7 tasks to 12 tasks, the time consumption increases dramatically from 319.1 seconds to 5621.9 seconds. For large-size instances, CPLEX solver is not only very time-consuming but also unavailable to output feasible solutions. Therefore, it is necessary to design effective methods so as to efficiently solve this problem.

5. Solution approaches

In this section, an easy-implement partition-and-sequencing task heuristic and a problem-based genetic algorithm are devised for effectively solving the problem.

5.1. Partition-and-sequencing task heuristic

5.1.1. Procedures

The main ideology of partition-and-sequencing task (PST) heuristic is to assign tasks to YCs in advance such that the YCs only process the assigned tasks independently. The main procedures of PST heuristic are presented as follows.

Step 1: Separate tasks into three segments from left to right and assign them to YCs 1, 2, and 3, respectively. Define $n = \lfloor \frac{|N|}{3} \rfloor$ where n is the minimum number of tasks assigned to each YC.

If $|N| = 3n$, the numbers of tasks assigned to the YCs are the same. For example, YCs 1, 2, and 3 are responsible for the tasks in slots 2 and 15, 27 and 42, 53 and 62, respectively (see Figure 3a). If $|N| = 3n + 1$, one YC has to process $n + 1$ tasks.

For example in Figure 3b, there are three assignment cases under the 7-task instance. Similarly, it can be observed that there also exist three assignment cases if $|N| = 3n+2$, i.e., the 8-task instance in Figure 3c.

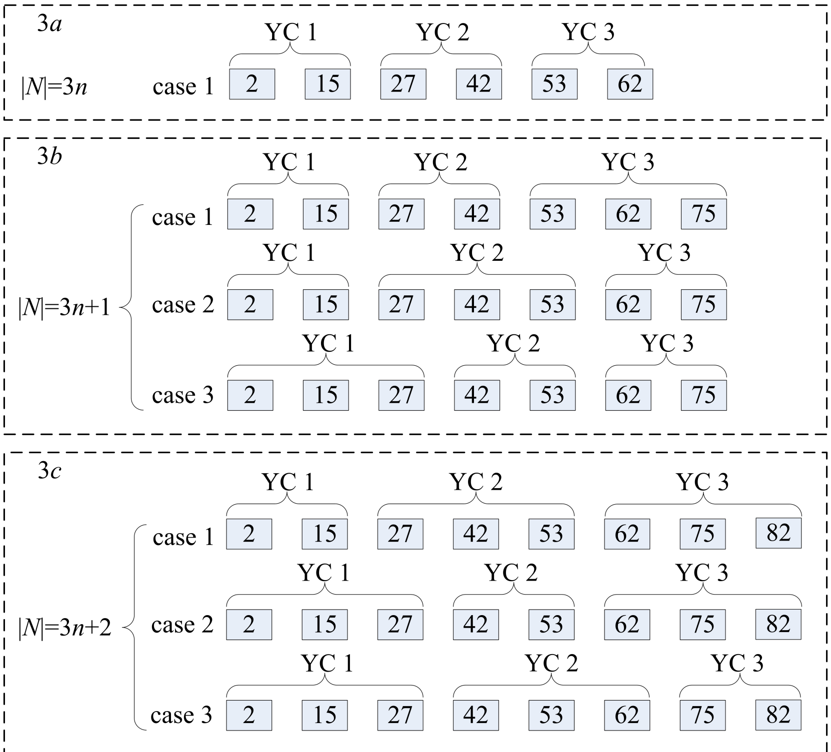

Fig. 3 Task partition under different number of tasks.

Step 2: For each case, sequence the tasks of YCk, $k = 1, 2, 3$, according to the non-decreasing order of their release times, i.e., a first-come-first-service ($FCFS$) rule.

Step 3: Conduct the following procedures based on the task sequence in Step 2.

(i) Before any YC starts processing a task, check whether the safety distance with its adjacent YCs (under processing) is respected. If not, the YC must wait for the completion of tasks on the adjacent YCs;

(ii) Update the layer of non-processed tasks after the completion of one task;

Step 4: Repeat Step 3 until all tasks have been processed; Compute the sum of deviation between c_i and r_i for all tasks (i.e., the objective value).

Step 5: Repeat Steps 2 to 4 until all three cases of the instance are computed. Select the case with the minimum objective value as the final output.

5.1.2. Computational complexity of PST heuristic

According to the above Steps 1 to 4, the computational complexity for each case is

$O(|N|)$ since the computational complexity is at most $O(n+1)$ for a YC. There is at most three cases of task assignment for any instance, thus the computational complexity for the proposed PST heuristic is $O(3|N|)$. Compared with CPLEX solver, the major merit of PST heuristic is its fast running speed for producing feasible solutions for the considered problem.

5.2. Genetic algorithm

The above PST heuristic processes tasks using a greedy thought, thus it may cause unexpected idle times of YCs. In this subsection, the genetic algorithm (GA) proposed by Goldberg and Holland (1988) is adopted to resolve the considered problem. The main idea of GA is to search for good feasible solutions by simulating natural evolutionary processes. GA begins with an initial population. In the next generations, individuals with higher fitness are selected as parents to produce adaptable offsprings. The procedures are repeated until the stopping criterion is met. The best individual in the last generation is set as the solution to the problem.

5.2.1. Encoding representation

In this work, a chromosome, with two equal-length segments, is used to represent a solution for the problem. The first segment reveals the processing sequence of $|N|$ tasks, while the second segment corresponds to the assignment of YCs. Figure 4 illustrates an example of the encoding representation of an individual. It is seen that the task sequence is 1, 6, 5, 3, 4, 2, 7 which are assigned to YC 3, 2, 1, 3, 2, 2, 1, respectively.

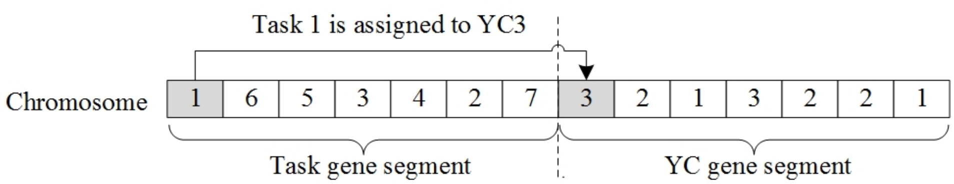

Fig. 4 An example of encoding representation.

5.2.2. Feasible solution generation

In order to guarantee the feasibility of each individual in the initial population, it is regulated that YCs 1 and 3 cannot process tasks in blocks 2 and 1, respectively. Besides, the tasks at the 8-leftmost (resp. right) slot of container block 1 (resp. 2) can only be served by YC1 (YC3) due to the inter-crane interference. The following Algorithm 1 describes the generation of feasible individuals for the initial population, where *pop* represents the population size. Its principle is to randomly generate the task assignment and eliminate the impossible cases, respecting the inter-crane interference.

Algorithm 1 Generation of initial feasible solutions (encoding procedure)

Input: $|N|, b_z^L, b_z^R, pop$.

Output: Feasible solutions.

```
chrom = ∅ (%Initial the chromosome matrix);
chrom ← (pop, 2|N|) (%Define the size of the chromosome matrix);
for j = 1 : pop do
    chrom(j, 1 : |N|) = randperm(|N|) (%Generate the task sequence randomly);
    chrom(j, |N| + 1 : 2|N|) = randi([1, 3], 1, |N|) (%Assign one YC randomly for each task);
    for i = 1 : N do
        if b_chrom(j, i) ≤ b_1^R & chrom(j, |N| + i) = 3 then
            chrom(j, |N| + i) = randi([1, 2], 1, 1)
            (%Regenerate a YC if task i in block 1 has been assigned to YC3);
        end if
        if b_chrom(j, i) ≥ b_2^L & chrom(j, |N| + i) = 1 then
            Chrom(j, |N| + i) = randi([2, 3], 1, 1)
            (%Regenerate a YC if task i in block 2 has been assigned to YC1);
        end if
        if b_chrom(j, i) ≤ 8 then
            chrom(j, |N| + i) = 1 (%Assign YC1 to the leftmost 8 slots);
        end if
        if b_chrom(j, i) ≥ b_2^R − 8 then
            chrom(j, |N| + i) = 3 (%Assign YC3 to the rightmost 8 slots);
        end if
    end for
    Update chrom.
end for
```

5.2.3. Decoding process and fitness evaluation

In this part, the feasible solutions obtained by Algorithm 1 is decoded such that its corresponding objective value can be computed. First, the start and completion times of each individual should be computed considering the inter-crane interference and position-dependent processing times by in Algorithm 2. The principle of Algorithm 2 is to identify the task location and verify whether the safety distance is respected before any YC's processing. If yes, this YC can process the task. For example, one probability is that the next task for YC2 locates at the left of on-processing YC1, thus YC2 should wait for the current task on YC1, then both YCs 1 and 2 adjust their locations towards the left side. After processing a task, calculate its processing time and update the locations of non-processed tasks. Algorithm 2 terminates until all tasks have been processed by YCs.

Based on the output of Algorithm 2, the objective values in one generation can be computed. Due to the minimization nature of the problem, the fitness function can be expressed as $Fitness = 1/ObjV$ to identify the fitness of each individual.

Algorithm 2 Calculate the start and completion times (decoding procedure)

Input: $|N|$, b_i, h_i, $chrom$.
Output: s_i, c_i.

$s_i, c_i = zeros(pop, |N|)$;
Initial the locations $(loc1, loc2, loc3)$ of three YCs according to Assumption (iii);
while $loc2 - loc1 \geq 8$ & $loc3 - loc2 \geq 8$ (%All YCs keep the safety distance now) **do**
  for $j = 1 : pop$ **do**
    for $i = 1 : |N|$ **do**
      $p\_chrom(j, i) = u[(2 \cdot h\_chrom(j, i) - 1]$; (%Calculate the processing time);
      if $b\_chrom(j, i) \leq 8$ & $chrom(j, |N| + i) = 1$ **then**
        (%Case 1: Tasks in the leftmost 8 slots);
        if $loc2 - b\_chrom(j, i) \geq 8$ **then**
          The safety distance is satisfied, YC1 begins to process task i; Compute s_i and c_i of task i; Renew $loc1$; Renew the locations of rest tasks;
        end if
      else
        if $loc2 - b\_chrom(j, i) < 8$ **then**
          $loc2 = loc1 + 8$ (%Ensure the safety distance constraint);
        end if
        if $loc1 = b\_chrom(j, i)$ **then**
          continue (%Judge the real-time location of YC1 and break loop in case 1);
        end if
      end if
      if $b\_chrom(j, i) > 8$ & $chrom(j, |N| + i) = 1$ & $b\_chrom(j, i) \leq loc1$ **then**
        (% Case 2: Task i locates at the left of YC1 and is assigned to YC1. Compute s_i and c_i of task i; Renew $loc1$; Renew the locations of rest tasks;
        if $loc1 = b\_chrom(j, i)$ **then**
          continue (%Judge the real-time location of YC1 and break loop in case 2);
        end if
      end if
      if $b\_chrom(j, i) > 8$ & $chrom(j, |N| + i) = 2$ & $b\_chrom(j, i) \leq loc1$ **then**
        (%Case 3: Task i locates at the left of YC1, however, it is assigned to YC2. Move YC1 to keep a safety distance; Compute s_i and c_i of task i; Renew $loc1$ and $loc2$; Renew the locations of rest tasks;
        if $loc2 = b\_chrom(j, i)$ **then**
          continue (%Judge the real-time location of YC2 and break loop in case 3);
        end if
      end if
      (%Here we just present 3 cases to illustrate the ideology).
    end for
  end for
end while

5.2.4. Genetic operator designs

To enlarge the diversification in each generation, the selection, crossover and mutation operators are applied to generate new individuals.

(i) Selection operator. Each individual in the current generation is assigned with a slice of a circular roulette wheel, in accordance with its fitness value. That is, the one with a higher fitness value takes more proportion on the wheel, indicating a higher opportunity of being selected as a parent. This *roulette wheel sampling* method is applied to construct a mating pool for further procedures.

(ii) Crossover operator. Two chromosomes are randomly selected from the mating pool. Then a real number between 0 and 1 is generated arbitrarily for the two chromosomes. If it is less than the given crossover probability, they perform a crossover procedure. Specifically, two points are randomly selected in the second segment $(|N|+1, 2|N|)$ for the selected individuals. The genes between these two points are exchanged, while the rest ones keep unchanged. Figure 5 illustrates a typical crossover procedure.

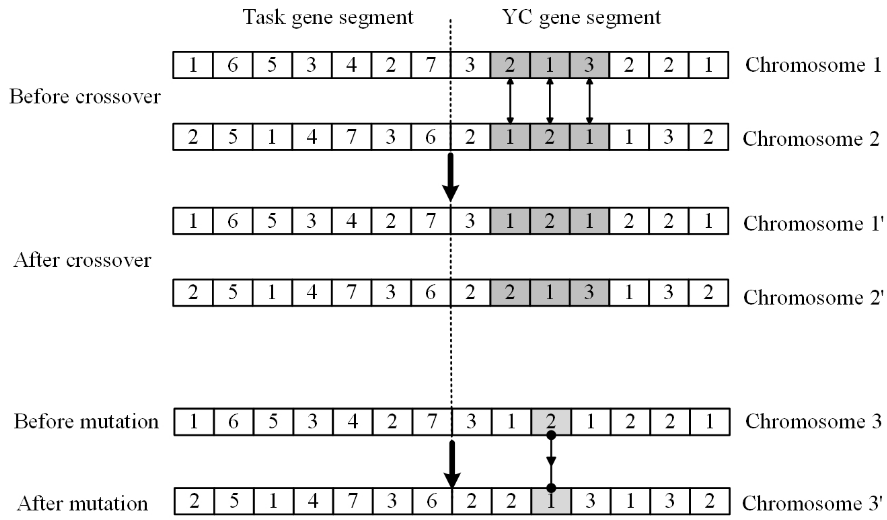

Fig. 5 An example of crossover and mutation procedures.

(iii) Mutation operator. For an arbitrary chromosome in the mating pool, a real number between 0 and 1 is randomly generated. Any chromosome who fails to exceed the given probability is selected to perform mutation procedure. One gene of a chromosome is selected in the second segment of a chromosome, i.e., in $(|N| + 1, 2|N|)$. It is replaced by a randomly generated value among $\{1, 2, 3\}$ different from the original YC index. Figure 5 also shows a typical mutation procedure.

5.2.5. Available test and new population insertion

After the above operators, newly generated offsprings that violate the safety distance constraint (infeasible) will be discarded. To keep the population size for the next iterations, newly feasible individuals are created based on the principle of Algorithm 1.

The feasible individual with the best objective value will be reserved after each iteration, referred to as an *elite reserved strategy*. The above procedures repeat in each iteration until the stop criterion is met. Finally, the individual with the minimum objective value in the last iteration is selected as the solution for the problem. In summary, the framework of the proposed GA is presented in Algorithm 3, where $Maxgen$, p_c and p_m represent to the maximum number of iterations, crossover probability, and mutation probability, respectively.

Algorithm 3 The framework of the proposed GA.

Input: $|N|, a_i, b_i, h_i, r_i, Maxgen, pop, p_c, p_m$.
Output: $\sum_{i \in N}(c_i - r_i)$.

Initialize $Chrom = zeros(pop, 2|N|)$, $trace = zeros(1, Maxgen)$ and set $gen = 0$;
Encode $Chrom$ by **Algorithm 1**;
Decode $Chrom$ by **Algorithm 2**; Calculate $ObjV$ for each chromosome and obtain its fitness;
while $gen < Maxgen$ **do**
  Rank the fitness of all chromosomes in $Chrom$;
  Select higher-fitness chromosomes (refer to as $Selch$) to the mating pool;
  Crossover procedure for $Selch$;
  Mutation procedure for $Selch$;
  Examine the feasibility of newly generated individuals;
  Decode $Selch$ by **Algorithm 2**;
  Calculated $ObjV$ for each chromosome;
  if the number of chromosomes in $Selch$ is less than pop **then**
    Insert feasible chromosomes into $Selch$ to satisfy pop; Record the minimal objective value;
  end if
  $gen = gen + 1$;
  $trace(1, gen) = min\{ObjV\}$; (% trace the minimum objective)
  if $gen == 1$ **then**
    $trace(1, gen) = min\{ObjV\}$; (% trace the minimum objective)
    $minVal = min\{ObjV\}$; (% record the minimum objective)
  end if
  if $gen \geq 1$ **then**
    $trace(1, gen) = min\{trace(1, gen-1), min\{ObjV\}\}$; (% identify the smaller objective)
  end if
end while
Output $minVal$ and set it as the objective value.

5.2.6. Computational complexity of GA

According to Algorithm 3, it is seen in each iteration that all the sub-algorithms are not in a nested loop. This demonstrates that the algorithm with the largest computational complexity determines that of GA. To be specific, Algorithms 1 and 2 have a computational complexity of $O(pop \cdot |N|)$, while crossover and mutation procedures are $O(pop)$. Computing the objective values has a computational complexity of $O(pop \cdot |N|^2)$, which is the largest one. Since there are a total of *gen* iterations. Therefore, the computational complexity of the proposed GA can be derived as $O(gen \cdot pop \cdot |N|^2)$.

6. Numerical experiments

In this section, numerical experiments are conducted to evaluate the performance of proposed algorithms. Both of PST heuristic and GA are programmed in Matlab 2014a and executed on a personal computer (Intel Core i5 with 4 GB RAM) with Microsoft Windows 7 operating system. Commercial solver CPLEX 12.6 is used and called in Matlab 2014a to solve the MIP model. The computational time of CPLEX is limited by 7200 seconds (two hours in CPU time) since large computational time is unacceptable for terminal practice.

6.1. Instance generation

The container block layout corresponds to the practice in Shanghai Port. For a container block, the maximum numbers of rows, slots, and layers are up to 6, 40, 6, respectively (refer to as Figure 1). Besides, the bounds of blocks 1 and 2 are set as $[b_1^L, b_1^R] = [1, 40]$ and $[b_2^L, b_2^R] = [47, 86]$. The minimum safety distance between any two YCs d is set as 8, and there is a distance $c = 6$ between two blocks.

The generated instances are classified into six sets, i.e., $S1$ and $S2$ for small-size instances, $M1$ and $M2$ for medium-size instances, and $L1$ and $L2$ for large-size instances. Each of which contains 10 instances with the same size. The numbers of tasks for three sizes are up to 12, 50, 100, respectively. The three-tuple location of a target task is randomly generated without exceeding the block size, and its release time is randomly generated within an interval $[0, 40]$.

For the MIP model, the planning horizon contains up to 1000 time steps for small-size instances. Without loss of generality, one time step is equal to 20 seconds in practice. The processing time of a single container movement $u = 3$ time steps, i.e., 60 seconds. Besides, the traveling time of a yard crane between two adjacent slots equals to 1 time step, i.e., 20 seconds.

The relevant parameters of the proposed GA are specified in the next part, using a turning method, including the population size, crossover probability, mutation probability, and the number of iterations.

6.2. GA Parameter turning

The number of iterations is set as 200 since all the instances tend to be stable before the 200th iteration according to preliminary experiments. In order to ensure the performance of the genetic algorithm, a three-step procedure (El Amraoui et al., 2013) is adapted to select a better combination of GA parameters. The parameter turning processes are shown as follows:

- **Step 1:** Choose 10 instances generated for testing the performance of GA under different parameter combinations.

- **Step 2:** Define 8 kinds of parameter combinations represented by $\Psi_1, \Psi_2, ..., \Psi_8$ (in Table 1). For the first instance, **(i)** choose combination Ψ_1, run the instance by GA for 10 times and record the average objective value and average computational time. **(ii)** Change the parameter combination and run the instance by GA similar to Step 2 (i), until all parameter combinations are examined. **(iii)** Change another instance and repeat Step 2 (i) and (ii), until all instances have been examined.

- **Step 3:** The Friedman test followed by the Bonferroni-Dunn test are applied for choosing the best combination. The former test decides whether the null hypothesis is rejected or not, if yes, the second test helps to decide the best combination. Please refer to Nguyen et al. (2012) and El Amraoui et al. (2013) for more details.

Table 1: Combination of GA parameters and the calculated mean ranks.

Combination	Ψ_1	Ψ_2	Ψ_3	Ψ_4	Ψ_5	Ψ_6	Ψ_7	Ψ_8
Parameter values								
pop size	50	50	50	50	100	100	100	**100**
p_c	0.7	0.7	0.8	0.8	0.7	0.7	0.8	**0.8**
p_m	0.2	0.3	0.2	0.3	0.2	0.3	0.2	**0.3**
Mean ranks $\overline{\Psi}_\omega$								
Cost	7.38	6.88	6.00	4.63	4.38	2.75	2.00	1.50
$Time(s)$	1.36	1.35	1.31	1.27	2.46	2.43	2.44	2.45
Deviation to $\overline{\Psi}_8$								
Cost	5.88	5.38	4.50	3.13	2.88	1.25	0.50	0
$Time(s)$	-1.09	-1.1	-1.14	-1.18	0.01	-0.02	-0.01	0

For each instance, the Friedman test ranks the 8 kinds of combinations, giving 1 to the best one and 8 to the worst. See Table 1, the indicator *cost* defines the means of the combination ranks, and *time* refers to the computational time (CPU time in seconds). Then, regarding the Bonferroni-Dunn test, the pairwise comparisons of mean ranks suggest that Ψ_8 is the best one, i.e., $pop = 100$, $p_c = 0.8$ and $p_m = 0.3$.

6.3. The computational result on a 7-task instance

In this part, a 7-task instance is used to illustrate the validity of the formulated MIP model. The input information of it is presented in Table 2, where the planning horizon $|T| = 100$. For example, task 1 occupies the location $(a_1, b_1, h_1) = (1, 4, 3)$ and its release time is 0.

Table 2: The input information of a 7-task instance.

Task index	1	2	3	4	5	6	7
a_i	1	3	2	6	3	4	5
b_i	4	52	20	53	52	70	82
h_i	3	2	4	2	1	4	5
r_i	0	20	7	4	10	6	11

The Gantt chart obtained by CPLEX for this instance is shown in Figure 6. Each shaded rectangle represents the processing of a task, and the two numbers under the rectangle denote the start and completion times. The tasks from the bottom row to the top row are processed by YCs 1, 2 and 3 respectively. The optimal objective value is 169 and the computational time is 283.5 seconds.

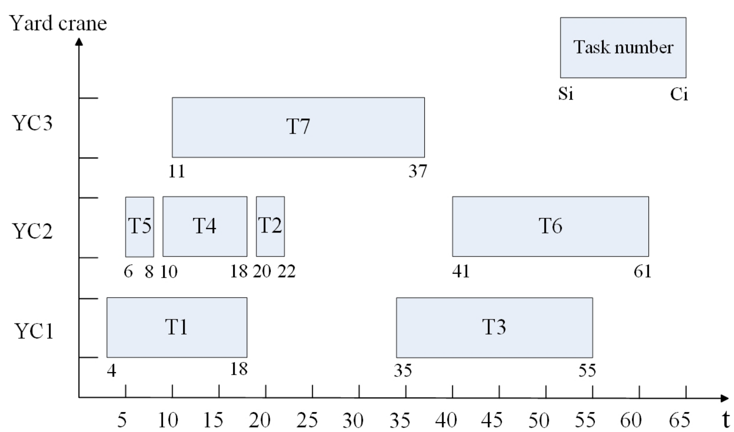

Fig. 6 The optimal schedule for the 7-task instance by CPLEX.

6.4. Result comparison and analysis

In this part, numerical experiments are conducted based on the instances generated in Section 6.1. The corresponding computational results are reported and analyzed in the following.

6.4.1. Comparison and analysis under small-size instances.

The computational results of small-size instances (i.e., $S1$ and $S2$) are reported in Table 3. The first column indicates the number of instances. For example, 7-1 means the first instance among the 7-task instances. Columns 2 and 3 record the objective values and computational time of CPLEX. In terms of GA, columns 4-6 present respectively the objective values, the computational time and the deviation Gap_{cg}, where $Gap\_cg = (Obj\_g - Obj\_c)/Obj\_c \times 100\%$. Similarly, columns 7-10 correspond to the results of PST heuristic where deviations $Gap\_cp = (Obj\_p - Obj\_c)/Obj\_c \times 100\%$ and $Gap\_gp = (Obj\_p - Obj\_g)/Obj\_g \times 100\%$. Besides, the average values of indicators are presented at the bottom of Table 3.

Table 3: Experiment comparison under small-size instances

Instance	CPLEX		Genetic Algorithm			PST Heuristic			
($S1$)	*Obj_c*	*time(s)*	*Obj_g*	*time(s)*	*Gap_cg(%)*	*Obj_p*	*time(s)*	*Gap_cp(%)*	*Gap_gp(%)*
7-1	169	283.5	169	2.45	0	190	0.0036	12.4	12.4
7-2	235	395.2	237	2.51	0.9	284	0.0025	20.9	19.8
7-3	144	221.8	144	2.52	0	151	0.0029	4.9	4.9
7-4	178	276.1	180	2.51	1.1	184	0.0028	3.4	2.2
7-5	204	304.3	204	2.50	0	297	0.0033	45.6	45.6
7-6	254	287.5	257	2.49	1.2	297	0.0035	16.9	15.6
7-7	217	335.2	217	2.53	0	375	0.0022	72.8	72.8
7-8	185	276.4	185	2.52	0	194	0.0035	4.9	4.9
7-9	**267**	287.9	**267**	2.48	0	**267**	0.0034	**0**	**0**
7-10	221	523.2	221	2.47	0	332	0.0034	50.2	50.2
Average	207.4	319.1	208.1	2.49	0.3	257.1	0.0031	23.2	22.8
($S2$)									
12-1	408	5448.8	431	4.25	5.6	527	0.0014	29.2	22.3
12-2	490	4490.2	518	4.24	5.7	816	0.0011	66.5	57.5
12-3	511	5230.5	523	4.22	2.3	701	0.0014	37.2	34.0
12-4	433	6939.2	447	4.20	3.2	537	0.0009	24.0	20.1
12-5	396	6293.7	411	4.24	3.8	475	0.0012	19.9	15.6
12-6	450	6591.2	456	4.21	1.3	686	0.0009	52.4	50.4
12-7	446	6345.2	461	4.23	3.3	585	0.0011	31.2	26.9
12-8	440	4662.6	453	4.21	3.0	587	0.0012	33.4	29.5
12-9	398	4976.3	417	4.21	4.8	602	0.0007	51.3	44.3
12-10	373	5241.3	388	4.25	4.0	472	0.0008	26.5	21.6
Average	434.5	5621.9	451.5	4.22	3.8	598.8	0.0011	37.2	32.6

For the instances in $S1$ (i.e., 7-task instances), it can be seen that the optimal objective values in column 2 range from 144 to 267 whose average is 207.4, and the average computational time is 319.1s by CPLEX. The GA can provide optimal solutions for 7 out of 10 instances, and the average gap *Gap_cg* is 0.3 that is very small as well. PST heuristic can propose optimal solutions (see instance 7-9) with a very small probability. However, the average gap *Gap_cp* is more than 20% on average. In terms of CPU time, both of PST heuristic and GA show a great superiority over CPLEX: PST and GA take only 0.003s and 2.49s, respectively, while CPLEX needs 319.1s on average, to get the optimal solution.

The computational results under $S2$ (12-task instances) are presented in the second part of Table 3. The objective values obtained by GA are very close to those of CPLEX, and the average gap is only 3.8%. However, the average gap between PST heuristic and CPLEX is 37.2%. In terms of running time, CPLEX consumes more than 5600 seconds which is a very sharp increase, compared with 7-task instances. Although CPLEX can produce optimal values, it becomes very time-consuming when the number of tasks has a small increase. On the contrary, PST heuristic and GA can solve the problem with average computational time only 0.001s and 4.22s, respectively.

From these small-size instances (i.e., sets $S1$ and $S2$), it can be concluded that the proposed GA not only provides near-optimal solutions but also saves a lot of computational time. Although PST heuristic can solve this problem with a shorter computational time, its average gaps with CPLEX and GA are very large. Note that PST heuristic takes less computational time on 12-task instances than that of 7-task instances. The reason is that the computational complexity for 12-task and 7-task instance are $O(|N|)$ and $O(3|N|)$, respectively (Section 5.1.2).

6.4.2. Comparison and analysis under medium-size and large-size instances.

As CPLEX cannot solve large scale instances within a short time due to NP-hard nature, in this part, PST heuristic and GA are evaluated by medium-size and large-size instances. For medium-size instances, the numbers of tasks for sets $M1$ and $M2$ are 30 and 50, respectively. While for large-size instances, the two sets $L1$ and $L2$ contain 70 and 100 tasks, respectively.

We can observe in Table 4 that, under the 30-task instances, the average objective values of GA and PST heuristic are respectively 4531.3 and 5605.9 and they have an average gap of 24%. For the 50-task instances, the average gap between them becomes 13.9%. In terms of the computational time, we can observe that both GA and PST can solve the problem within a very short time: GA can solve the 30-task and 50-task instances by 20.9s and 63.4s on average, and PST heuristic only uses 0.006s and 0.027s for solving the medium-size instances.

Table 4: Experiment comparison under medium-size and large-size instances

Instance	Genetic Algorithm		PST Heuristic			Instance	Genetic Algorithm		PST Heuristic		
($M1$)	*Obj_g*	*time(s)*	*Obj_p*	*time(s)*	*Gap_gp*(%)	($M2$)	*Obj_g*	*time(s)*	*Obj_p*	*time(s)*	*Gap_gp*(%)
30-1	4523	21.2	6066	0.0064	34.1	50-1	13086	65.1	15120	0.029	15.5
30-2	4324	20.8	5772	0.0066	33.5	50-2	11921	63.2	12557	0.027	5.3
30-3	4145	20.6	5149	0.0062	24.2	50-3	12987	62.9	14706	0.029	13.2
30-4	4198	21.0	4982	0.0061	18.7	50-4	13376	64.3	14491	0.028	8.3
30-5	4896	20.9	5266	0.0069	7.6	50-5	12489	62.9	15612	0.027	25.0
30-6	4863	20.9	5987	0.0056	23.1	50-6	13573	64.8	15725	0.026	15.9
30-7	4097	20.5	5708	0.0065	39.3	50-7	13321	61.7	14383	0.028	8.0
30-8	4493	21.0	5278	0.0066	17.4	50-8	13006	64.0	14247	0.027	9.5
30-9	5291	21.1	6431	0.0059	23.9	50-9	13181	62.7	15579	0.026	18.2
30-10	4583	20.7	5420	0.0065	18.3	50-10	13476	62.4	16105	0.025	19.5
Average	4531.3	20.9	5605.9	0.0063	24.0	*Average*	13041.6	63.4	14853	0.027	13.9
($L1$)						($L2$)					
70-1	24487	130.5	25372	0.061	3.6	100-1	52545	300.9	56947	0.072	8.4
70-2	25888	132.4	28116	0.054	8.6	100-2	54910	302.8	57780	0.068	5.2
70-3	23343	131.8	24816	0.053	6.3	100-3	50029	301.4	54101	0.065	8.1
70-4	28069	129.2	30461	0.051	8.5	100-4	54738	303.7	55003	0.069	0.5
70-5	26085	131.7	28814	0.057	10.5	100-5	55654	301.4	59553	0.062	7.0
70-6	26836	129.3	27512	0.054	2.5	100-6	55923	300.4	57903	0.071	3.5
70-7	25037	130.4	27561	0.058	10.1	100-7	54432	303.1	59061	0.066	8.5
70-8	27543	131.5	30763	0.059	11.7	100-8	54837	302.5	57260	0.064	4.4
70-9	24241	130.7	25869	0.052	6.7	100-9	57702	301.6	62006	0.062	7.5
70-10	26430	131.1	28012	0.052	6.0	100-10	55799	303.2	58893	0.067	5.5
Average	25795.9	130.1	27730	0.055	7.4	*Average*	54656.9	302.1	57851	0.067	5.9

As for the large-size instances in Table 4 (i.e., $L1$ and $L2$), the computational time of GA is still acceptable with the increase of the number of tasks. The GA can solve the 100-task problems with around 5 minutes. In contrast, PST can still solve the 100-task instances immediately, with an average of 0.067s. Besides, it is seen in Table 4 that the average gaps between GA and PST heuristic are 7.4% and 5.9% for sets $L1$ and $L2$, respectively. This trend indicates that if the instance size is large enough, the two heuristics GA and PST will generate a closer performance on solving this complex scheduling problem.

We can conclude from the experimental results that CPLEX can provide optimal solutions for small-size instances. The GA can produce high-quality solutions within an acceptable time. PST heuristic can provide solutions quickly, but the solution qualities are inferior to those proposed by GA. In summary, CPLEX using MIP model and the proposed GA are suggested to be used to help relieve port congestion phenomenon.

7. Conclusions

This paper investigates a three-yard-crane scheduling problem in two container blocks in a line, considering the safety distance, the inter-crane interference and position-dependent processing times. In order to improve the efficiency of container terminals, a MIP model is proposed to minimize the total flow time of tasks. Due to NP-hard complexity of the problem, it is impossible to obtain optimal solutions within a reasonable time for large-size applications. Therefore, two heuristics PST and GA are devised to effectively solve the considered problem. Numerical experiments validate the MIP model and demonstrate the efficiency of our proposed heuristics, which show incremental contribution in daily operations for container terminals.

Further research directions may contain: (i) Improve the proposed genetic algorithm such that some infeasible solutions can be repaired in an efficient way. (ii) Consider the synergistic scheduling of yard cranes within the different kinds of container blocks in line. (iii) Uncertainties (such as urgent demands) should be considered in multiple container blocks. (iv) Consider the periodic maintenance and random breakdown of the yard cranes in order to guarantee the service of target containers.

References

[1] Bierwirth C., F. Meisel. A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research*, 2010, 244(202): 615-627.

[2] Cao J., D. H. Lee, J. Chen, Q. Shi. The integrated yard truck and yard crane scheduling problem: Benders decomposition-based methods. *Transportation Research Part E*, 2010, 46(3): 344-353.

[3] Carlo H. J., I. F. A. Vis, K. J. Roodbergen. Storage yard operations in container terminals: Literature overview, trends, and research directions. *Flexible Service and Manufacturing Journal*, 2014, 235(2): 412-430.

[4] Chen G., K. Govindan, Z. Yang, T. M. Choi, L. Jiang. Terminal appointment system design by non-stationary $M(t)/E_k/c(t)$ queueing model and genetic algorithm. *International Journal of Production Economics*, 2013, 146(2): 694-703.

[5] Do N. A. D., I. E. Nielsen, G. Chen, P. Nielsen. A simulation-based genetic algorithm approach for reducing emissions from import container pick-up operation at container terminal. *Annals of Operations Research*, 2016, 242(2): 285-301.

[6] Dolgui A., V. Gordon, V. Strusevich. Single machine scheduling with precedence constraints and positionally dependent processing times. *Computers & Operations Research*, 2012, 39(6): 1218-1224.

[7] El Amraoui A., M. A. Manier, A. El Moudni, M. Benrejeb. A genetic algorithm approach for a single hoist scheduling problem with time windows constraints. *Engineering Applications of Artificial Intelligence*, 2012, 26(7): 1761-1771.

[8] Gharehgozli A. H., G. Laporte, Y. Yu, R. Koster. Scheduling twin yard cranes in a container block. *Transportation Science*, 2015, 49(3): 686-705.

[9] Goldberg D. E., J. H. Holland. Genetic algorithms and machine learning. *Machine Learning* 1988, 3(2-3): 95-99.

[10] Guinet A. Scheduling sequence dependent jobs on identical parallel machines to minimize completion time criteria. *International Journal of Production Research*, 2007, 31(7): 1579-1594.

[11] Hsu C. J., D. Yang. Unrelated parallel-machine scheduling with position-dependent deteriorating jobs and resource-dependent processing time. *Optimization Letters*, 2014, 8(2): 519-531.

[12] Huang S. Y., X. Guo. Reducing simulation costs of embedded simulation in yard crane dispatching in container terminals. *ACM Sigsim Conference on Principles of Advanced Discrete Simulation*, Canada, May 19-22, 2013, 305-314.

[13] Huang S. Y., Y. Li, X. Guo. Yard crane dispatching to minimize total weighted vessel turnaround times in container terminals. *Springer International Publishing*, 2015: 1747-1758.

[14] Jian G., D. H. Lee, J. Cao. Storage yard management in maritime container terminals. *Transportation Science*, 2014, 50(4): 1300-1313.

[15] Li W., Y. Wu, M. E. H. Petering, M. Goh, R. Souza. Discrete time model and algorithms for container yard crane scheduling. *European Journal of Operational Research*, 2009, 198: 165-172.

[16] Liu M., F. Chu, Z. Zhang, C. Chu. A polynomial-time heuristic for the quay crane double-cycling problem with internal-reshuffling operations. *Transportation Research Part E*, 2015a, 81: 52-74.

[17] Liu M., F. Zheng, J. Li. Scheduling small number of quay cranes with non-interference constraint. *Optimization Letters*, 2015b, 9(2): 403-412.

[18] Liu M., C. Y. Lee, Z. Zhang, C. Chu. Bi-objective optimization for the container terminal integrated planning. *Transportation Research Part B*, 2016, 93: 720-749.

[19] Ng W. C., K. L. Mak. An effective heuristic for scheduling a yard crane to handle jobs with different ready times. *Engineering Optimization*, 2005, 37(8): 867-877.

[20] Ng W. C. Crane scheduling in container yards with inter-crane interference. *European Journal of Operational Research*, 2005, 164(1): 64-78.

[21] Nguyen V. P., C. Prins, C. Prodhon. A multi-start iterated local search with tabu list and path relinking for the two-echelon location-routing problem. *Engineering Applications of Artificial Intelligence*, 2012, 25(1): 56-71.

[22] UNCTAD, 2016. Review of maritime transport: 2015. ISBN: 9789211128109.

[23] Vis I. F. A., H. J. Carlo. Sequencing two cooperating automated stacking cranes in a container terminal. *Transportation Science*, 2010, 44(44): 169-182.

[24] Wu Y., W. Li, M. E. H. Petering, M. Goh, R. Souza. Scheduling multiple yard cranes with crane interference and safety distance requirement. *Transportation Science*, 2015, 49(4): 1-16.

[25] Xu D., D. Yang. Makespan minimization for two parallel machines scheduling with a periodic availability constraint: Mathematical programming model, average-case analysis, and anomalies. *Applied Mathematical Modelling*, 2013, 37(14-15): 7561-7567.

[26] Yin Y., W. Wu, T. C. E. Cheng, C. Wu. Single-machine scheduling with time-dependent and position-dependent deteriorating jobs. *International Journal of Computer Integrated Manufacturing*, 2014, 28(7): 1-10.

[27] Yin Y., Y. Wang, T. C. E. Cheng. Parallel-machine scheduling of deteriorating jobs with potential machine disruptions. *Omega*, 2017, 69: 17-28.

[28] Zhang C., T. Wu, K. H. Kim, L. Miao. Conservative allocation models for outbound containers in container terminals. Parallel-machine scheduling of deteriorating jobs with potential machine disruptions. *European Journal of Operational Research*, 2014, 238(1): 155-165.

[29] Zhen L., E. P. Chew, L. H. Lee. An integrated model for berth template and yard template planning in transshipment hubs. *Transportation Science*, 2011, 45(4): 483-504.

[30] Zhen L. Storage allocation in transshipment hubs under uncertainties. *International Journal of Production Research*, 2016, 52(1): 72-88.

[31] Zhen L., Z. Xu, K. Wang, Y. Ding. Multi-period yard template planning in container terminals. *Transportation Research Part B*, 2016, 93: 700-719.

[32] Zheng F., X. Man, F. Chu, M. Liu, C. Chu. Two yard crane scheduling with dynamic processing time and interference. *IEEE Transactions on Intelligent Transportation Systems*, 2018a, 19(12): 3775-3784.

[33] Zheng F., Y. Li, F. Chu, M. Liu, Y. Xu. Integrated berth allocation and quay crane assignment with maintenance activities. *International Journal of Production Research*, 2018b, doi: 10.1080/00207543.2018.1539265.

[34] Zhu Z., L. Sun, F. Chu, M. Liu. Single-machine group scheduling with resource allocation and learning effect. *Computers & Industrial Engineering*, 2012, 60(1): 148-157.