A stochastic beam search for the berth allocation problem

Fan Wang a,⁎, Andrew Lim b

a School of Business, Sun-Yat Sen University, Guangzhou 51075, China
b Department of Industrial Engineering and Logistics Management, The Hong Kong University of Science and Technology, Hong Kong, China

Received 12 April 2005; received in revised form 12 June 2006; accepted 13 June 2006
Available online 20 July 2006

Abstract

In this paper, the optimization of the Berth Allocation Problem (BAP) is transformed into a multiple stage decision making procedure and a new multiple stage search method, namely stochastic beam search algorithm, is proposed to solve it. New techniques such as an improved beam search scheme, a two-phase node goodness estimation, and a stochastic node selection criteria are proposed. Real-life information provided by Singapore Port was collected as our test data. Experimental results show that the proposed stochastic beam search is more accurate and efficient than both the state-of-the-art meta-heuristic and the traditional determinist beam search.

© 2006 Elsevier B.V. All rights reserved.

Keywords: Port transportation; Allocation; Beam search

1. Introduction

The rapid increasing number of container shipments is causing higher demands on the seaport container terminals, container logistics, management and technical equipment. From the statistics of 2002 World’s top ten container ports [1] listed in Table 1, the volume (in Twenty feet Equivalent Units (TEU)) increased over 10% in average from 2001 to 2002. For example, in China, the TEU volume increased 35% in Shanghai port and 50% in Shenzhen port. Compared with the TEU volume, the Gross Domestic Product (GDP) of China only increased 8% from 2001 to 2002. Therefore, the marine container industry has grown dramatically in the past years.

Handling the world second largest port, Port Singapore Authority (PSA) operates four seamlessly inter-connected container terminals – Tanjong Pagar, Keppel, Brani and Pasir Panjang Terminals which handle over 40,000 containers and 60 vessels calls on an average day. It provides vessels a choice of 200 shipping lines with connections to 600 ports in 123 countries including daily sailing to every major port in the world [26].

Port authorities always attempt to provide cost optimization and service enhancements. A seaport successes with short berthing duration for vessels and low prices for loading and discharging. Furthermore, minimizing the berthing duration for vessels becomes an overall objective with respect to terminal operations. The terminal operations are complex. Generally, the terminal operation system consists of three subsystems: vessel planning, storage and stacking, and transportation. Based on the monthly planned or on-line schedule of
arrival vessels, a container vessel is assigned to a berth wharf equipped with cranes to load and unload containers. Then, unloaded import containers are transported to a yard. On the other hand, export containers arriving by road or railway at the terminal are picked up by the internal equipment and distributed to the respective stocks in the yard. Trucks and rails move within the terminal for containers storage and transportation.

In this paper, we study the first operation in vessel planning, namely “Berth Allocation Problem (BAP)”.

BAP determines the position and time for each arrival vessel to minimize the total berthing cost, where configurations, estimated arriving time and operation durations for vessels are known in advance. We model the berth wharf by a vertical line with a fixed length and model each vessel by a rectangle where the vertical side represents its overall-length and the horizontal side represents its operation duration. Hence, BAP becomes a constraint two-dimensional unrotation bin packing problem (see Fig. 1).

In the literature, BAP has been studied since the 1990s. Brown et al. used an integer programming model for assigning one possible berthing position to a vessel on various practical constraints [4]. They considered a berth as a collection of discrete berthing positions and berth shifting of moored vessels was allowed. Lim minimized the berth length while keeping the vessels’ berthing time fixed [19]. He transformed the problem into a restricted form of the two-dimensional packing problem and explored a graphical theoretical representation with a heuristic solution. Furthermore, Chia et al. presented an Ants Colony Optimization heuristic for minimizing the berth length [5]. Actually, BAP for minimizing the berth length is similar to the off-line Dynamic Storage Allocation (DSA) problem which is NP-hard. Approximation factors for DSA were found as 6 [15], 5 [7] and 3 [8]. When the wharf length is fixed as a constraint, Nishimura et al. proposed a Genetic Algorithm (GA) for dynamic berth assignment to vessels in the public berth system [22]. However, their result is not suitable for container ports. Imai et al. assumed that each vessel occupied exactly one berth and presented a mix-integer programming formulation with its Lagrangian relaxation [13]. In addition, a GA-based heuristic with different service priorities was proposed by Imai et al. [14]. Kim and Moon created a mix-integer programming to determine berthing time and position.
and proposed a Simulated Annealing (SA) heuristic [16]. Park and Kim applied a subgradient optimization method [24]. Park and Kim combined the berth assignment approach with the quay crane scheduling [25]. In simulation study, Legato and Mazza addressed a queuing network model and a simulation experiment of vessels arrival, berthing and departure at a container terminal [17]. Recently, Dai et al. proposed a state-of-the-art SA heuristic for BAP where they reported that their SA outperformed other published methods [6]. The search space and neighborhood of their SA is based on the concept of sequence pair for the two-dimensional rectangle packing problem [12].

BAP becomes a set of machine scheduling problems when we assume the arriving time of all vessels are zero. Li et al. addressed the more general problem of BAP, namely “scheduling with multiple-job-on-one-processor pattern” by minimizing the make-span of the schedule [18]. They derived several approximation bounds for a variant of the first-fit decreasing heuristic. Guan et al. studied the case with weighted completion time and proposed a heuristic solution [10]. Later, Guan et al. solved this general problem for large size input by a tree-search heuristic [9]. However, since they ignored the arriving time of vessels, the above results cannot be directly applied to BAP in practice.

BAP is one of the core components in the port decision support system for both off-line and on-line process. However, BAP is NP-hard and finding effective bounds is also difficult. In this paper, we propose a new heuristic search, namely Stochastic Beam Search, and apply it to solve the BAP. The stochastic beam search consists of a new beam search framework, a two-phase node goodness estimation, and a stochastic node selection (sampling) criteria. Computational results clearly show that our proposed stochastic beam search outperforms both the current state-of-the-art meta-heuristic and the traditional beam search.

The rest paper is organized as follows. Section 2 discusses the framework of the port decision support system and indicates the requirements for BAP solution methods. We stated the BAP in Section 3 by a mixed-integer programming model and transform it into a multi-stage decision making procedure by Dynamic Programming. Section 4 proposes our new framework of the stochastic beam search with its major components. Then, we implemented the proposed stochastic beam search for solving the BAP in Section 5. Section 6 introduces Dai et al.’s SA in brief for comparison. Computational experiments on real-life data are demonstrated and analyzed in Section 7. Finally, we concluded the paper in Section 8.

2. Decision support system

We demonstrate the framework of decision support system for vessel planning in Fig. 2. Data input to the system consist both berth terminal data and vessel data. Berth terminal data includes berth configurations such as the berth terminal wharf length and the berth terminal manpower configuration. On the other hand, there are two types of vessel data, monthly vessel plan and online.
vessel schedule. Vessel data include the vessel essential information (Identification, Overall-length and Estimated arrival time), operation information (Operation time duration and Manpower requirement), cost information (Unallocation cost, Berthing position cost and Delay berthing cost). The system consists of four subsystems: BAP, online BAP, crane split, and stowage planning. The BAP subsystem first processes a pre-planning (off-line planning) to generate a berthing plan for crane split and stowage planning subsystems, based on monthly vessel plan and berth terminal data. In addition, subject to the uncertainty of vessel schedule, the online BAP vessel plan and berth terminal data. In addition, subject to the uncertainty of vessel schedule, the online BAP, online BAP, crane split, and stowage planning.

The BAP subsystem first processes a pre-planning (off-line planning) to generate a berthing plan for crane split and stowage planning subsystems, based on monthly line planing) to generate a berthing plan for crane split and stowage planning subsystems, based on monthly vessel plan and berth terminal data. In addition, subject to the uncertainty of vessel schedule, the online BAP, online BAP, crane split, and stowage planning.

To build an efficient BAP optimization engine for implementing a high-quality decision support system for vessel planning, the following criteria and missions need to be considered:

1. Accurate optimum on the total cost;
2. Fast response, especially for the on-line process;
3. Easy implementation and maintenance for variants on objective function and constraint;
4. Concise and uniform structure for high-quality team-work coding and testing.

3. Problem statement

Notations.

L: the length of the berth wharf.

Vessels \( \{v_i | 1 \leq i \leq N\} \) are sequenced in increasing order of their arrival time, where \( N \) is the number of vessels. For each vessel:

- \( l_i \): length-overall including the requested gap between adjacent vessels for safety;
- \( a_i \): estimated arrival time;
- \( d_i \): time duration of operation;

Decision variables:

\[ y_i: y_i = 1 \text{ if vessel } v_i \text{ is allocated; otherwise } y_i = 0; \]
\[ x_i: \text{ allocation position; } \]
\[ t_i: \text{ allocation time. } \]

Total costs:

\[ C_0: \text{ unallocation cost, } C_0 = c_0(1 - y_i), \]
\[ C_1: \text{ position cost, } C_1 = c_1(x_i - q_i)^+, \]
\[ C_2: \text{ delay cost, } C_2 = c_2(t_i - a_i), \]

BAP can be classified by the variants on cost configuration as BAP\((c_0; c_1; c_2)\). We formulate BAP by the following mixed-integer programming model.

\[
\min \sum_{i=1}^{N} c_0(1 - y_i) + y_i(c_1(x_i - q_i)^+ + c_2(t_i - a_i))
\]

subject to

\[
\begin{align*}
x_i + l_i & \leq L, \quad \forall y_i = 1, \quad 1 \leq i \leq N \\
t_i & \geq a_i, \quad \forall y_i = 1, \quad 1 \leq i \leq N \\
x_i + l_i & \leq x_j \text{ or } x_j + l_j \leq x_i \quad \text{or} \\
t_i + d_i & \leq t_j \text{ or } t_j + d_j \leq t_i, \\
\forall y_i = y_j = 1, \quad & \forall i, j \leq N \\
y_i \in \{0, 1\}; & \quad t_i \leq 0, \quad \forall 1 \leq i \leq N
\end{align*}
\]

Eq. (1) indicates the objective function. Eq. (2) guarantees that the position of each allocated vessel does not exceed the wharf length. Eq. (3) means that allocating cannot be operated before arriving for each vessel. Eq. (4) implies no overlap between vessels. The decision variables are indicated in Eq. (5).

The status of a vessel \( v_i \) is remarked by a triple \((y_i, x_i, t_i)\). Therefore, an allocation plan can be represented by a vector of all \( N \) vessels’ statuses, i.e., \( (y_1, x_1, t_1), (y_2, x_2, t_2), \ldots, (y_N, x_N, t_N) \). Hence, BAP can be transformed into a multi-stage decision-making procedure with \( N \) steps where one vessel is considered at each step.

Then, Dynamic Programming (DP) is applied to model the above multi-stage decision-making procedure. A state in the DP is defined as \( S((y_1, x_1, t_1), (y_2, x_2, t_2), \ldots, (y_N, x_i, t_i)) \) which represents the partial solution for the first \( i \) vessels. Hence, the DP model can be formulated as follows where the initial state is \( S(\emptyset) = 0 \).

\[
S((y_1, x_1, t_1), (y_2, x_2, t_2), \ldots, (y_1, x_1, t_1)) = \min_{\forall \text{feasible}(y, x, t)} S((y_1, x_1, t_1), (y_2, x_2, t_2), \ldots, (y_{i-1}, x_{i-1}, t_{i-1}) + c_0(1 - y_i) + y_i(c_1(x_i - q_i)^+ + c_2(t_i - a_i))
\]

Definition 1. Wharf-adjacent set is a set of vessels whose horizontal sides (allocation positions) are connected with each other.
Definition 2. Time-adjacent set is a set of vessels whose vertical sides (arriving time) are connected with each other.

For instance, in the allocation plan illustrated in Fig. 3, vessels v1 and v3 are in the same wharf-adjacent set. While v1 and v2 are in the same time-adjacent set.

Proposition 1. In an optimal solution of BAP,

(1) for any wharf-adjacent set \( V^w \) (\( V^w \subseteq V \)), either of the following two conditions holds:
   
   (a) at least one vessel in \( V^w \) is located at its least-cost berthing position, e.g. \( \exists v_j \in V^w, x_j = q_j \).
   
   (b) at least one vessel in \( V^w \) is located on the right or left boundary of the wharf, e.g. \( \exists v_j \in V^w, x_j = 0 \) or \( x_j = L - l_j \).

(2) for any time-adjacent set \( V^t \) (\( V^t \subseteq V \)), at least one vessel in \( V^w \) is berthed at its estimated arrival time, e.g. \( \exists v_k \in V^t, t_k = a_k \).

The Proof of Proposition 1 is simple since we can shift the wharf-adjacent set along the vertical direction as well as shift the time-adjacent set along the horizontal direction until the total cost defined by Eq. (1) cannot no longer be reduced.

4. Stochastic beam search

There are a number of search strategies in artificial intelligence for problem solving. Breadth First Search (BFS) expands the shallowest node in the search tree first. In contrast, Depth First Search (DFS) expands the deepest node in the search tree first. If the estimated remaining distance from any current node to the goal node is close to and always smaller than the true remaining distance, A* search can find the optimum without too much effort. If we just evaluate the goodness of the current node by historical information, A* search is degenerated into a B&B search. B&B always expands the shortest path leading into an open node until there is a path reaching the goal that is of length no longer than all incomplete paths terminating at open nodes.

However, for solving NP-hard problems, we prefer to obtain near-optimal approximations for large sized data instead of the optimal solutions for only small sized data. Different to other heuristic search such as local search, tabu search, GA and SA, beam search is a breadth-first style heuristic search and progresses along with the depth step by step. Unlike BFS, beam search only expands nodes that are likely to succeed at each level. Only these nodes are kept in the beam, the rest are pruned to improve efficiency. In literature, beam search was first applied to speech recognition systems [20]. Ow and Morton first studied the performance behavior of beam search with other heuristics for scheduling problems and proposed a beam search implementation with high quality solutions for the single machine early/tardy problem and the flow shop weighted tardiness problem [23]. Nair et al. developed a beam search for the product line design problem [21]. Recently, Beraldi and Ruszcynski applied beam search to solve stochastic integer problems under probabilistic constraints [2].

We illustrate the idea of our stochastic beam search in Fig. 4. The stochastic beam search starts from Set \( B^1 \) (e.g., the set of \( b_1 \) initial nodes in the first level (“Level 1”)). It generates nodes Set \( R^2 \) with \( r_2 \) nodes by the neighborhood operator from Level 1 to Level 2. In Level 2,

Step 1 \( U^2 \) is constructed as a subset of \( R^2 \) by a draft selection process;

Step 2 \( B^2 \) is then constructed as a subset of \( U^2 \) by a detail selection process;

Step 3 All nodes in \( B^2 \) are expanded to construct the Set \( R^3 \) for the next level (“Level 3”).

The above estimation–selection–expansion procedure is repeated level by level until it reaches the final level (“Level \( N \)”). All nodes in Set \( R^N \) are feasible solutions and the best solution among them is returned as the final best solution. The size of search space is \( \sum_{i=1}^{N} r_i \). If the beam width \( r_i \) is fixed to \( r \), the computational complexity of stochastic beam search is \( rN \). To implement a stochastic beam search, we need to develop the following seven components:

(1) Level: the whole decision-making procedure is divided into \( N \) levels and the solution is constructed by expanding the nodes level by level;

(2) Node: there are three types of nodes – nodes generated and to be expanded (Set B), nodes generated (Set R), and nodes pruned;
(3) Neighborhood Operators: to generate new nodes in the next level from the nodes in the current level;
(4) Beam width \( r_i = |R_i| \): the number of nodes that are generated in each level \( i \);
(5) Filtered beam width for the draft selection \( u_i = |U_i| \);
(6) Filtered beam width for the detail selection \( b_i = |B_i| \);
(7) Goodness estimation of nodes for the draft selection and the detail selection to construct Set \( U \) and Set \( B \).

Our stochastic beam search aims to improve the performance of the traditional beam search scheme, motivated by:

(1) increasing the diversity of nodes in each level;
(2) seeking effective goodness estimation methods to construct Set \( B \) in each level, trading-off between the accuracy and time complexity.

4.1. Node goodness estimation

The goodness estimation function \( f \) is the sum of historical and future parts, i.e., \( f = g + h \) where \( f \) is the global estimation, \( g \) is the pre-estimation only from Level 1 to the current level, and \( h \) is the post-estimation only from the current level to the final level. Since \( g \) is historical, the computation complexity of pre-estimation is trivial. In contrast, to make the post-estimation more accurate, obtaining \( h \) is time-consuming in practice. Therefore, we proposed a two-phase goodness estimation and selection to balance the accuracy and computational complexity, “draft selection” first where \( f = g + h_1 \) and “detail selection”

Fig. 4. Framework of beam search.
second where \( f = g + h_2 \), where the computational complexity for calculating \( h_1 \) is much smaller than that of calculating \( h_2 \). The \( h_1 \) based draft selection first rapidly selects potential nodes from \( R \) to \( U \) to reduce the input size for the detail selection. Then, \( h_2 \) is calculated for each node in \( U \) and Set \( B \) is constructed by the detail selection.

4.2. Stochastic selection

\( m \) potential nodes are selected from \( M \) candidate nodes based on their goodness estimation results, where \((m, M) = (u_i, r_i)\) for the draft selection and \((m, M) = (b_i, u_i)\) for the detail selection. In all published beam search implementations, “top \( m \) method” are always applied as the selection criteria, which ranks all \( M \) candidate nodes in increasing order of their \( f \) and selects the top \( m \) nodes while pruning the remaining \( M - m \) nodes.

In contrast, we propose a new “stochastic method” as the selection criteria to improve diversities between nodes. Since the goodness estimation results are possibly inaccurate, “stochastic method” can help the beam search escape from the wrong search directions. Different to the “top \( m \) method”, the “stochastic method” gives opportunities to all \( M \) candidate nodes, not just to the top \( m \) nodes in the ranking list. Moreover, the possibility of selection for each candidate node depends on its rank. The nodes in higher ranks have the more possibilities to be selected, i.e., \( p(n_1) > p(n_2) \) if \( f(n_1) < f(n_2) \). Hence, the probability function is a monotonic increasing function.

There are many ways to define a probability function. To simplify, we applied a piecewise probability function in our beam search implementation for solving the BAP that demonstrated in Fig. 5. According to the rank list of candidate nodes in increasing order of their \( f \), the set of all \( M \) candidate nodes (denoted as Set \( S \)) is partitioned to the three disjoint subsets, e.g., the set of the first \( s + 1 \)-th node to the \( 3s \)-th node (denoted as “Set \( S^1 \)”), and the set of the remaining \( M - 3s \) nodes (denoted as “Set \( S^3 \)”). We assign a probability of \( \frac{5}{6s + M} \) to each node in Set \( S^1 \), \( \frac{2}{6s + M} \) to each node in Set \( S^2 \), and \( \frac{1}{6s + M} \) to each node in Set \( S^3 \). The following equation formalizes the piecewise probability function where \( n \) denotes a candidate node and \( p(n) \) represents the probability of selection assigned to \( n \).

\[
    p(n) = \begin{cases} 
    \frac{5}{6s + M} & \text{if } n \in S^1, \\
    \frac{2}{6s + M} & \text{if } n \in S^2, \\
    \frac{1}{6s + M} & \text{if } n \in S^3. 
    \end{cases}
\]

Hence, \( \sum_{n \in S} p(n) = 1 \).

Stochastic selection, also namely stochastic sampling, for node selection is a well-known idea to improve the diversity of solutions during search and therefore to obtain a better solution than that by determinist sampling based search [3]. For example, a probabilistic greedy search algorithm is proposed to solve set covering problem and the experimental results show that the stochastic sampling based diversification schemes improves the robustness and quality performance of the greedy search [11]. For beam search, the traditional “top \( m \)” selection method is a kind of determinist sampling. To our best knowledge, we are the first to develop a beam search based on stochastic sampling. In general, stochastic sampling for node selection is more suitable for beam search than that for other search methods because beam search need to keep large diversity of nodes in each search level to obtain accurate solutions finally.

5. Implementation for solving the BAP

The status of a vessel \( v_k \) (\( 1 \leq k \leq N \)) is remarked by a triple \((y_k, x_k, t_k)\). Therefore, an allocation plan can be represented by a vector of all \( N \) vessels’ statuses, i.e., \((y_1, x_1, t_1); (y_2, x_2, t_2), \ldots, (y_N, x_N, t_N)\). Hence, BAP can be transformed into a multi-stage decision-making procedure with \( N \) levels where one vessel is considered at each level.

We implement the seven components of stochastic beam search as follows:

1. Level: there are \( N \) levels in total and we allocate one vessel in each level in increasing order of vessels’ arrival time \( a_k \).
2. Node: a node in level \( k \) is a vector \((y_1, x_1, t_1); (y_2, x_2, t_2), \ldots, (y_k, x_k, t_k)\).
3. Neighborhood operator: to construct \((y_{k+1}, x_{k+1}, t_{k+1})\) from any node in \(B^k\), in the allocation position axis, we assume that \(v_{k+1}\) is either in the same wharf-adjacent set with previous allocated vessels or \(y_{k+1}\) connected with the left or the right boundaries of the wharf. Initially, \(v_1\) can be placed at any position. On the other hand, in time axis, we enumerate \(t_{k+1}\) from \(a_{k+1}\) to \(a_{k+1} + T_{\text{max}}\) for each vessel \(v_{k+1}\), where \(T_{\text{max}}\) is the maximum delay threshold. Fig. 6 illustrates the neighborhood for allocating \(v_5\) (areas 1–6) based on the positions of the first four vessels. The neighborhood for allocating \(v_6\) depends on the position of \(v_5\).

4. Beam width \(r_i = +\infty\);

5. Filter beam width for the draft selection \(u_i = u\) where \(u\) is a constant;

6. Filter beam width for the detail selection \(b_1 = b = \frac{1}{3}u\);

7. Goodness estimation: DFS estimation We set \(h_1 = 0\) and \(h_2\) represents the minimum cost in the next \(\Delta\) levels. In other words, the estimation for the draft selection is the same as the pre-estimation where the goodness of nodes are ranked by their historical performances only, i.e., \(f = g\). In contrast, in our detail selection \((f = g + h_2)\), we define \(h_2\) is the minimum cost of the next \(\Delta\) vessels only by a DFS algorithm, instead of by searching all remaining \(N-k\) levels.

For the special case, if we set \(\Delta = N-k\), we can have \(f = g + h_2 = g + h^* = f^*\) where \(h^*\) represents the actual minimum cost for all remaining vessels in the future for the current partial solution. However, since DFS is time-consuming due to the NP-hardness of the BAP, \(\Delta\) must be set as a small constant.

6. Compared heuristic

Dai et al. proposed a state-of-the-art SA meta-heuristic for solving the BAP. Their experimental results show that the SA meta-heuristics out-performs other

---

**Table 2**

List of tunable parameters in experiments

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Meaning</th>
<th>Value in experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td>(r)</td>
<td>Beam width</td>
<td>(+\infty)</td>
</tr>
<tr>
<td>(u)</td>
<td>Filtered beam width for draft selection</td>
<td>1000</td>
</tr>
<tr>
<td>(b)</td>
<td>Filtered beam width for detail selection</td>
<td>33, 66, 150, 330</td>
</tr>
<tr>
<td>(T_{\text{max}})</td>
<td>Maximum delay threshold</td>
<td>60</td>
</tr>
<tr>
<td>(\Delta)</td>
<td>Search depth for DFS based detail selection</td>
<td>3, 4, 5, 6</td>
</tr>
<tr>
<td>(s)</td>
<td>Parameter of node set (S) participation</td>
<td>(b)</td>
</tr>
</tbody>
</table>

---

For the special case, if we set \(\Delta = N-k\), we can have \(f = g + h_2 = g + h^* = f^*\) where \(h^*\) represents the actual minimum cost for all remaining vessels in the future for the current partial solution. However, since DFS is time-consuming due to the NP-hardness of the BAP, \(\Delta\) must be set as a small constant.

---

**Fig. 6.** Illustration of neighborhood construction in beam search for solving the BAP.
published methods for solving BAP in terms of accuracy and efficiency. In their SA heuristic, the search space is encoded by the concept of “Sequence Pair” [6]. Each allocation plan then is transformed to a pair of permutations of all vessels \((H, V)\). The neighborhood of search space is constructed by the following four operators:

1. Single Swap: interchange two vessels’ positions in either sequence \(H\) or sequence \(V\);
2. Double Swap: interchange two vessels’ positions in both \(H\) and \(V\) sequences;
3. Single Shift: select two vessels and slide one vessel along sequence \(H\) or sequence \(V\), until their relative positions are changed;
4. Double Shift: shift along both \(H\) and \(V\) sequences;

7. Computational results

Our computational experiments used real-life data from Singapore Port Terminal. All 40 test instances were classified into the four sets by the number of vessels – 10 instances for Small Set \((N \leq 100)\), 5 instances for Medium Set \((100 < N \leq 200)\), 19 instances for Large Set \((200 < N \leq 300)\) and 6 instances for Huge Set \((300 < N \leq 400)\). We implemented our proposed stochastic beam search by C/C++ programming language and compared it with the state-of-the-art SA meta-heuristic for solving the BAP. The computation results were obtained on an Intel Pentium IV – 1 GHz Personal Computer with 512 MB of RAM.

All tunable parameters for implementing the stochastic beam search are completely listed in Table 2.

We demonstrate the computational results in Table 3 to compare our proposed stochastic beam search with both Dai et al.’s SA meta-heuristic and the traditional beam search with “top \(m\)” node selection in the total cost and running time (in s). Because of the stochastic node selection, the costs and running time of stochastic beam search in Table 3 are averaged values by five runs of the beam search (the deviation of running time is obviously minor and that of costs is within 5% for the same instance). From the results, we have:

1. The proposed stochastic beam search outperformed the state-of-the-art SA meta-heuristic significantly for all 40 test instances, obtaining better results in a short running time. For example, for instance 185 in Medium Size, beam search obtained cost = 2 in 67 s while the heuristic obtained cost = 24 in 262 s. Another example is beam search finds cost = 25 in 772 s while the heuristic finds cost = 84 in 2213 s, for instance 172 of Huge Size. In addition, beam search found optimums for six instances 181, 182, 184, 186 and 189.

2. Although the proposed stochastic node selection leads minor extra computation time (from Table 3, the running time of stochastic beam search is a little bit bigger than that of traditional beam search), it helps to reduce the costs significantly for all 40 test instances. The above experimental results indicate that the stochastic node selection is more effective and robust than the determinist

<table>
<thead>
<tr>
<th>Instance</th>
<th>Stochastic beam search</th>
<th>Traditional beam search</th>
<th>SA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cost</td>
<td>Time</td>
<td>Cost</td>
</tr>
<tr>
<td>001</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>002</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>003</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>004</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>005</td>
<td>0</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>101</td>
<td>0</td>
<td>27</td>
<td>3</td>
</tr>
<tr>
<td>102</td>
<td>0</td>
<td>37</td>
<td>2</td>
</tr>
<tr>
<td>103</td>
<td>1</td>
<td>37</td>
<td>3</td>
</tr>
<tr>
<td>104</td>
<td>0</td>
<td>24</td>
<td>5</td>
</tr>
<tr>
<td>105</td>
<td>0</td>
<td>19</td>
<td>6</td>
</tr>
<tr>
<td>181</td>
<td>0</td>
<td>56</td>
<td>10</td>
</tr>
<tr>
<td>182</td>
<td>0</td>
<td>42</td>
<td>12</td>
</tr>
<tr>
<td>183</td>
<td>3</td>
<td>48</td>
<td>6</td>
</tr>
<tr>
<td>184</td>
<td>0</td>
<td>36</td>
<td>2</td>
</tr>
<tr>
<td>185</td>
<td>2</td>
<td>67</td>
<td>8</td>
</tr>
<tr>
<td>144</td>
<td>6</td>
<td>1140</td>
<td>9</td>
</tr>
<tr>
<td>145</td>
<td>7</td>
<td>1144</td>
<td>8</td>
</tr>
<tr>
<td>161</td>
<td>9</td>
<td>606</td>
<td>9</td>
</tr>
<tr>
<td>165</td>
<td>11</td>
<td>490</td>
<td>13</td>
</tr>
<tr>
<td>186</td>
<td>0</td>
<td>64</td>
<td>1</td>
</tr>
<tr>
<td>187</td>
<td>3</td>
<td>52</td>
<td>2</td>
</tr>
<tr>
<td>188</td>
<td>1</td>
<td>58</td>
<td>1</td>
</tr>
<tr>
<td>189</td>
<td>0</td>
<td>77</td>
<td>0</td>
</tr>
<tr>
<td>190</td>
<td>11</td>
<td>178</td>
<td>11</td>
</tr>
<tr>
<td>191</td>
<td>14</td>
<td>160</td>
<td>15</td>
</tr>
<tr>
<td>192</td>
<td>9</td>
<td>92</td>
<td>23</td>
</tr>
<tr>
<td>193</td>
<td>6</td>
<td>106</td>
<td>8</td>
</tr>
<tr>
<td>194</td>
<td>6</td>
<td>109</td>
<td>9</td>
</tr>
<tr>
<td>195</td>
<td>9</td>
<td>130</td>
<td>19</td>
</tr>
<tr>
<td>196</td>
<td>10</td>
<td>150</td>
<td>12</td>
</tr>
<tr>
<td>197</td>
<td>13</td>
<td>143</td>
<td>16</td>
</tr>
<tr>
<td>198</td>
<td>6</td>
<td>164</td>
<td>10</td>
</tr>
<tr>
<td>199</td>
<td>14</td>
<td>189</td>
<td>24</td>
</tr>
<tr>
<td>200</td>
<td>12</td>
<td>170</td>
<td>13</td>
</tr>
<tr>
<td>146</td>
<td>19</td>
<td>1914</td>
<td>20</td>
</tr>
<tr>
<td>151</td>
<td>25</td>
<td>3523</td>
<td>26</td>
</tr>
<tr>
<td>153</td>
<td>36</td>
<td>3261</td>
<td>40</td>
</tr>
<tr>
<td>154</td>
<td>28</td>
<td>3097</td>
<td>31</td>
</tr>
<tr>
<td>167</td>
<td>20</td>
<td>668</td>
<td>23</td>
</tr>
<tr>
<td>172</td>
<td>25</td>
<td>772</td>
<td>29</td>
</tr>
</tbody>
</table>
top $m$ node selection for solving BAP by beam search in equivalent computation time.

In addition, we also tested the performance of our stochastic beam search in term of parameters setting for solving the BAP. We observed two important parameters. One is the filtered beam width for the detail selection, $b$, which indicates the number of nodes stored in each level. The other is the depth of the DFS, $\Delta$, which implies the accuracy of the DFS goodness estimation and highly impacts the running time. We first demonstrate the total cost vs. the filtered beam width for the detail selection in Fig. 7 where two configurations of $b$, $b=33$ and $b=330$ are compared. In general, the larger beam width leads to the better accuracy but definitely results in much more running time. Because our beam search is stochastic, there are few exceptions where the smaller beam width outperforms the larger beam width.

Second, we demonstrate the results on the total cost vs. the depth of DFS in Fig. 8. We compared the two configurations on $\Delta$: $\Delta=3$ and $\Delta=5$. The latter outperforms the former for Small Set, Middle Set and Large set. For Huge Set, the performance on accuracy of the two configurations are very similar because the depths $\Delta=3$ and $\Delta=5$ are both not sufficient to accurately predict the node goodness.

8. Conclusion

Motivated by the decision support system for port terminal operation, in this paper, we have proposed a new stochastic beam search scheme for solving the Berth Allocation Problem (BAP) and compared it with the state-of-the-art meta-heuristic. An improved beam search framework, a two-phrase node goodness estimation and a scholastic node section criteria improve the performance of the beam search. Since BAP is NP-hard
and it is difficult to find effective bounds or accurate estimations on node goodness, beam search has remarkable advantages for solving the BAP. In summary, the advantages of our proposed stochastic beam search are (1) accurate; (2) fast; (3) easy to modify and reuse when customers often request to revise the objective function; (4) easy to implement, test, track and tune the parameters. Definitively, the proposed stochastic beam search can be directly applied to solving other multi-stage decision making problems.

References