THE CROSS-ENTROPY METHOD FOR COMBINATORIAL OPTIMIZATION PROBLEMS OF SEAPORT LOGISTICS TERMINAL
Turkay Yildiz1, Funda Yercan2
1Dokuz Eylul University, Haydar Aliyev Bulv, N:32, 35340, Inciralti, Izmir, Turkey
2Yasar University, Dept of International Logistics Management, Izmir, Turkey
E-mails: 1turkayyildiz@iyte.edu.tr; 2funda.yercan@yasar.edu.tr
Received 21 February 2010; accepted 15 November 2010

Abstract. Studies on seaport operations emphasize the fact that the numbers of resources utilized at seaport terminals add a multitude of complexities to dynamic optimization problems. In such dynamic environments, there has been a need for solving each complex operational problem to increase service efficiency and to improve seaport competitiveness. This paper states the key problems of seaport logistics and proposes an innovative cross-entropy (CE) algorithm for solving the complex problems of combinatorial seaport logistics. Computational results exhibit that the CE algorithm is an efficient, convenient and applicable stochastic method for solving the optimization problems of seaport logistics operations.

Keywords: cross entropy algorithm, seaport terminal, logistics, metaheuristic, optimization, stochastic method.

1. Introduction

Logistics is a set of full activities that properly arrange all aspects of the manufacturing and distribution process to ensure the delivery of the right products to the right markets at the right time. Key logistics strategies for improved competitiveness range from the operational level of an effectively managing mutually connected group of processes to the strategic level of designing a flexible transfer and transportation system that allows for quick optimal routing, mode choice changes and decision-making processes. In terms of logistics activities and strategies with respect to the improved competitiveness, seaport logistics operations possess comparable characteristics. However, seaport logistics operations are constrained by close-fitting space layouts and some exceptional handling equipment with supplies of the sizeable numbers of containers and/or bulk cargo traffic with demands for superior logistics service. Therefore, achieving optimum and quick solutions to the problems of seaport logistics operations are crucial for the improved logistics service output.

Seaport logistics operations are divided mainly into three sections and embrace seaside, yard and landside operations each of which engages multiple joined processes. Within these processes, in order to have efficient logistics service output, strategic and operational bodies exist and vigorously necessitate optimal resource management solutions based on the changing parameters of operating conditions.

General optimization needs at seaport terminals are arranged mainly as berth allocation, crane assignment, crane scheduling, yard management, yard traffic management, workforce planning, sheltering/warehousing, hinterland operations and infrastructure connections, i.e. intermodal connections. In addition, one of the most important challenges to system optimization is the trustworthiness and performance of the algorithm. In dynamically managed terminal operations (e.g. Automated Guided Vehicles (AGVs), Automated Straddle Carriers (ASCs), Automated Lifting Vehicles (ALVs)), instant decisions play crucial roles in the entire terminal operations (Lokuge and Alahakoon 2004, 2007; Van Hee and Wijbrands 1998; Liu et al. 2002; Rashidi 2006). On the other side, in dynamically changing operating environments such as additional amounts of vessels waiting for service at the queue and intensified container traffic in the yard area, computing time required for specific decisions should not go beyond feasible time range set by as a default. Dynamically routing yard trailers (and/or ASCs, ALVs, AGVs, etc.) for particular locations require highly organized terminal systems. The optimization of these operations involve sophisticated planning of input and output parameters and stating optimization problems, thus leading a way to more complicated and long definitions of the optimization problem having more constraints and variables. In such situations, algorithm performance and its global search within the overall conditions for optimal values is highly important.
Many combinatorial optimization problems are NP-hard (Garey and Johnson 1979). To tackle with these problems, the researches devoted many efforts. Metaheuristics (i.e. the CE) is a systematic approach to obtain knowledge when searching for an algorithm. Therefore, the algorithm provides a better knowledge of the future search for a better solution.

This paper proposes a cross-entropy (CE) approach to solving seaport terminal problems. CE is a modern and innovative metaheuristic method introduced by R. Y. Rubinstein in 1997 (Rubinstein 1997). The method transforms the deterministic problem into the stochastic one and then uses rare event simulation techniques to solve the problem. The method involves an iterative procedure of two stages. Based on a specified mechanism, it first generates a random data sample and then updates the parameters of the random mechanism based on data to produce a better sample for the next iteration (Rubinstein 1999). Recently, the cross-entropy method has been receiving a great deal of attention from researchers as this method has an ability to deal effectively with combinatorial optimization problems. This method has also been successfully applied to complicated combinatorial optimization problems.

The paper includes a few sections. In the following section (section 2), general background information regarding optimization problems is given indicating the details of the cross-entropy method and algorithm. In section 3, fundamental seaport operation problems are exhibited imposing designated constraints. In section 4, the scenario based shortest tour problem, including data on randomly assigned nodes is solved applying the CE method. In section 5, the fundamental reason for choosing the CE method of solving seaport problems is briefly given. The paper is summarized in section 6.

2. Combinatorial Optimization (CO) and the CE algorithm

2.1. A Background: Combinatorial Optimization and Heuristic Algorithms

A combinatorial optimization problem can be expressed as:

\[ x^* = \min_{x \in D \subseteq X} f(x), \]

where: the objective is to find \( x^* \in D \subseteq X \). \( X \) is bounded by finite space and \( D \subseteq X \) is the subspace of feasible solutions. \( f : X \rightarrow R^1 \) is the objective function.

To obtain solutions for the types of problems as shown in (1), several approaches exist (Aarts and Korst 1989; Colomni et al. 1996; Dorigo et al. 1999; Goldberg 1989; Kim et al. 2004; Kim 2005; Kozan and Preston 1999; Lee and Chen 2009; Lee et al. 2005; Legato and Mazza 2001). Depending on the type of the solution, three main types of algorithms such as exact, heuristic and approximate can be represented in a characteristic manner (Sergienko et al. 2009):

- The return of the optimal solution in finite space is assured of exact algorithms. If the algorithm cannot solve the problem, an optimal solution will not present. On the other hand, exact algorithms cannot constantly be used to solve some variations of CO problems (e.g. dynamic problems and problems facing a lack of clarity).
- Heuristic algorithms in many cases can provide one of the ways of obtaining an optimal solution in a reasonable period and are usually the algorithms suggesting absent or unknown accuracy estimates.
- Approximate algorithms (evolutionary algorithms, swarm algorithms, stochastic local search, etc.) are often based on some heuristics and if existing, return a substitute solution to finite time and the preciseness of these solutions can be estimated.

Briefly, evolutionary algorithms originate from biological evolution such as the Genetic Algorithm (GA), the Memetic Algorithm (MA), etc. Swarm intelligence algorithms take advantage of a special technique used for identifying the local interaction of scent or swarms like Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), etc. A stochastic approach to local search algorithms, including Simulated Annealing (SA), exploits the development of a local search and then employs the best solution from the neighbourhood along with the worst value of the objective function.

2.2. The Cross-Entropy (CE) Method

The CE method was developed in 1997 by Rubinstein and has been adapted for combinatorial optimization solutions (Rubinstein 1997, 1999, 2001; Rubinstein and Kroese 2010; Rubinstein and Melamed 1998; Rubinstein and Shapiro 1993). The idea behind the CE method is to model an effective learning technique throughout the search process of the algorithm to solve combinatorial optimization problems. The method first produces a random sample from a pre-specified probability distribution function and then treats the sample to adjust the parameters of probability distribution in order to generate a better sample in the next iteration. The stochastic optimization problem is solved by identifying optimal importance sampling (IS) density that minimizes Kullback-Leibler (KL) distance regarding the original density function. KL distance is cross entropy between the original density function and the importance sampling density function. Distance \( D(g,h) \) is determined as a particular suitable criterion between densities \( g \) and \( h \).

\[ D(g,h) = E_g \ln \frac{g(x)}{h(x)} = \int g(x) \ln g(x)dx - \int g(x) \ln h(x)dx. \]

Alternatively, for discrete variables, the Kullback-Leibler (KL) divergence of \( Q \) from \( P \) can be depicted as:

\[ D_{KL}(P,Q) = \sum_x P(x) \log\left(\frac{P(x)}{Q(x)}\right); \]
\[
D_{KL}(P,Q) = - \sum_x P(x) \log(Q(x)) + \sum_x P(x) \log(P(x)),
\]
where: \( D_{KL}(P,Q) = H(P,Q) - H(P) \), \( H(P,Q) \) is cross-entropy between \( P \) and \( Q \), \( H(P) \) is the entropy of \( P \). The minimization of \( KL \) distance (cross-entropy) provides a definition for the parameters of density functions and generations of the enhanced feasible vectors. The method aborts when comes together into a solution to the feasible region.

General 0–1 integer maximization problem \((P)\) can be defined as:
\[
(P): \quad z^* = \max_{x \in X} f(x),
\]
where: \( X \subseteq B^n \) represents the feasible region. The CE method associates a stochastic estimation problem with \((P)\). Random vector \( X = (X_1, \ldots, X_n) \sim \text{Ber}(u) \) and the parameterized vector of \( v \) is \( u \). Density function \( \phi \) on \( X \) is parameterized by vector \( u \in [0,1]^n \). Consequently, the Bernoulli density function under the following probability density function (PDF) is:
\[
\phi(x,u) = \prod_{i=1}^{n} (u_i)^{x_i}(1-u_i)^{1-x_i},
\]
and the stochastic estimation problem (EP) is:
\[
(EP): \quad P_u(f(x) \geq z) = \sum_{x \in X} I_{f(x) \geq z} \phi(x,u),
\]
where: \( P_u \) is the probability measure value based on the given threshold \( z \) value where \( X \) values are drawn from distribution \( \phi(\cdot,u) \). The stochastic problem (SP) of interest where \( f(x) \) is greater or equal to some real number \( z \) in the probability of \( \phi(x,u) \) is:
\[
(SP): \quad I = P_u(f(x) \geq z) = \sum_{x \in X} I_{f(x) \geq z} \phi(x,u),
\]
Low probability (e.g. \( 10^{-5} \)) of \( I = P_u(f(x) \geq z) \) is called a rare event. \( I_{f(x) \geq z} \) is indicator function taking two values 1 or 0 based on the threshold value of \( z \):
\[
I_{f(x) \geq z} = \begin{cases} 1, & f(x) \geq z \\ 0, & \text{otherwise}. \end{cases}
\]
The unbiased estimator of \( I \) obtained by drawing random sample \( X_1, \ldots, X_N \) from probability distribution function (pdf) \( \phi(\cdot,u) \) using crude Monte-Carlo (cMC) simulation is:
\[
\hat{I} = \frac{1}{N} \sum_{i=1}^{N} I_{f(X_i) \geq z},
\]
where: a plain definition of cMC is drawing from the distribution of \( s \) having \( m \) samples as such \( s_1, s_2, \ldots, s_m \); thus, the estimate of \( E(f) \) is:
\[
\hat{E}_m(f) = \frac{1}{m} (f(s_1) + f(s_2) + f(s_3) + \ldots + f(s_m)).
\]
Therefore, with the probability of 1, \( \hat{E}_m(f) \) equals to \( E_m(f) \) as such,
\[
\lim_{m \to \infty} \hat{E}_m(f) = E_m(f).
\]
Error component \( \varepsilon \) is calculated as:
\[
\varepsilon = \hat{E}_m(f) - E_m(f).
\]
The expected value of \( E_{\phi}(f) \) over distribution \( \phi \) is defined as:
\[
E_{\phi}(f) = \int f(s) \phi(s) ds,
\]
when the value of \( f \) is non-zero with minor probability to generate a satisfactory result in terms of relative error \( \varepsilon \). The cMC method necessitates large numbers of samples. To continue with (10), in case of a rare event (e.g. \( 10^{-5} \)), situations for \( f(x) \) greater than 1 cMC can raise some acute problems. Thus, as an alternative, random sample \( X_1, \ldots, X_N \) from importance sampling (IS) can be taken along with density \( \theta \) on \( X \):
\[
\hat{I} = \frac{1}{N} \sum_{i=1}^{N} I_{f(X_i) \geq z} \frac{\phi(X_i,u)}{\theta(X_i)}.
\]
Thus, the expected value of new estimate \( E_{\phi}(f) \) is:
\[
E_{\phi}(f) = \int f(s) \frac{\phi(s)}{\theta(s)} ds = \int \frac{f(s)\phi(s)}{\theta(s)} ds = E_{\phi}(f).
\]
In (11), \( \hat{I} \) is the likelihood ratio (LR) or importance sampling (IS) estimator.

Reference vector (p) is estimated by:
\[
\hat{p} = \arg \max_p \frac{1}{N} \sum_{i=1}^{N} I_{f(X_i) \geq z} \ln \phi(X_i, p),
\]
and the solution of reference vector \( \hat{p} \) is obtained taking partial differentiation with respect to \( p_j \):
\[
\frac{\partial}{\partial p_j} \frac{1}{N} \sum_{i=1}^{N} I_{f(X_i) \geq z} \ln \phi(X_i, p) = 0,
\]
which gives the optimal updating rule:
\[
\hat{p}_j = \frac{\sum_{i=1}^{N} I_{f(X_i) \geq z} X_{ij}}{\sum_{i=1}^{N} I_{f(X_i) \geq z}}, \quad j = 1, \ldots, n.
\]
The objective of the algorithm is to increase \( z \) threshold values in each iteration \((z^0, z^1, \ldots)\) and then converge \( z \) into a value near global optimum or global optimum value \( z^* \). With the initial \( p^0 \) vector at each iteration \( \tau \), a new value of \( z^\tau \) involves the creation of new \( p^{\tau+1} \) vector used to draw sample population to generate \( z^{\tau+1} \). At each iteration, better \( p \) vectors \((p^0, p^1, \ldots)\) are created and each of those are used to generate better \( z \) values. The algorithm will stop when \( z \) converges to global optimum value \( z^* \).
3. About Some Pure Base Optimization Problems

Human beings constantly make decisions by adopting an optimizing behaviour as a desire for performing the given task in the best possible way with respect to some unique criterion to minimize costs or maximize benefits (Ehrgott and Gandibleux 2002) as this is the case for any seaport operations, including organizational and process level activities. Seaport logistics operations have numbers of problems and mainly embrace the categories of scheduling, assignment, routing, allocation, shortest tour etc. Particular problems are ranging from scheduling problems of quay cranes, yard cranes, workforce and trailer and vehicle routing problems to the layout problems of sheltering, storage, warehousing operation etc. The fundamental problems of optimization applicable to the seaport terminal are briefly described and remodelled (Sarker and Newton 2007) to address seaport logistics operations:

- **Quay crane/yard crane scheduling** is the scheduling problem with an assumption that there are \( n \) jobs and \( m \) machines. Each job must be processed on all machines (i.e., cranes) in the given order. A machine (i.e., crane) can only process one job at a time and once the job is started on any of the machines (i.e., crane), it must be processed to completion. The objective is to minimize the sum of completion times of all jobs.

Objective function:

\[
\text{Minimize } Z = \sum_{j=1}^{n} t_{j(m),j},
\]

subject to:

\[
\begin{align*}
& t_{j(r+1),j} \geq t_{j(r),j} + P_{j(r),j} \\
& \text{for } r = 1, 2, ..., m-1 \text{ and } \forall j \\
& t_{ij} - t_{ik} \leq -P_{ik} + U(1 - x_{ijk}) \quad \forall i, j, k \\
& t_{ik} - t_{ij} \leq -P_{ij} + Ux_{ijk} \quad \forall i, j, k \\
& t_{ij} \geq 0 \quad \forall i, j \\
& x_{ijk} \in \{0, 1\} \quad \forall i, j, k,
\end{align*}
\]

where parameters are: \( n \) – the number of jobs; \( m \) – the number of machines; \( P_{ij} \) – the processing time of job \( j \) on machine \( i \); \( j(r) \) – the order of machines/operations for job \( j \) (for example, job \( j \) must be processed on machine 2 first \( r = 1, i = 2 \), then machine 4 \( r = 2, i = 4 \), etc.), for any job \( j, r = m \) means the last operation of the job, and variables: \( t_{ij} \) – the start time of job \( j \) on machine \( i \); \( x_{ijk} = 1 \) if job \( j \) precedes job \( k \) on machine \( i \), 0 otherwise (i.e., if job \( k \) precedes job \( j \) on machine \( i \)). More detailed studies and fundamental features of the applied framework on crane scheduling are published by Bierwirth and Meisel (2009); Chen et al. (2007); Goodchild and Daganzo (2007); Kim and Park (2004); Lee et al. (2008a, 2008b); Liang et al. (2008); Lim et al. (2002, 2004, 2007); Liu et al. (2006); Peterkofsky and Daganzo (1990); Tavakkoli-Moghaddam et al. (2009); Zhu and Lim (2006).

- **Scheduling (employees, stevedore, etc.)** is the problem that determines the number of employees required to meet different daily work force necessities of a seaport terminal thus minimizing the general scheduling cost.

Objective function:

\[
\text{Minimize } Z = \sum_{i=1}^{N} C_i x_i,
\]

subject to:

\[
\begin{align*}
& \sum_{i \in M_j} x_i \geq R_j \quad \forall j \\
& x_i \geq 0 \quad \forall i,
\end{align*}
\]

where parameters are: \( N \) – the total number of roster type; \( M_j \) – the set of roster types that will allow working on day \( j \); \( R_j \) – the number of employees required on each day \( j \); \( C_i \) – weekly cost per an employee assigned to roster type \( i \), and variables: \( x_i \) – the number of employees assigned to roster type \( i \).

For more detailed studies and fundamental features of the applied framework on scheduling refer to Li et al. (1998); Pinedo (2008).

- **Routing problem at seaport terminals** is the problem of ascertaining the operation plan satisfying demand for various zones at minimum cost.

Objective function is:

\[
\text{Minimize } f_{\text{obj}} = \sum_{i=1}^{G} \sum_{j=1}^{Z} \sum_{k=1}^{F} C_{ijk} x_{ijk},
\]

subject to:

\[
\begin{align*}
& \sum_{i=1}^{G} \sum_{k=1}^{F} L_{ik} x_{ijk} \geq D_j \quad \forall j \\
& \sum_{j=1}^{Z} \sum_{k=1}^{F} L_{ijk} x_{ijk} \leq S_j \quad \forall i \\
& \sum_{j=1}^{Z} L_{ijk} x_{ijk} \leq U_{ki} \quad \forall k, i \\
& x_{ijk} \geq 0 \quad \forall i, j, k,
\end{align*}
\]

where parameters are: \( G \) – the number of source locations (index \( i \)); \( Z \) – the number of receiving nodes for containers (index \( j \)); \( F \) – the number of the trailers available (index \( k \)); \( L_{ik} \) – load capacity of trailer \( k \); \( S_{j} \) – the quantity of the available containers for transportation from location \( i \); \( D_{j} \) – the quantity of containers required by zone \( j \); \( C_{ijk} \) – the unit cost of transporting from location \( i \) to zone \( j \) by trailer \( k \); \( U_{ik} \) – maximum allowable containers that can be transported from location \( i \) by trailer \( k \) in a given period, and variables: \( x_{ijk} \) – the number of trips required by trailer \( k \) from location \( i \) to zone \( j \).
For other detailed studies and fundamental features of the applied framework refer to Bish et al. (2001); Kim and Bae (1998); Vis and De Koster (2003).

- **Hinterland operations and landside operations (Vehicle Routing Problem):** a generic model that practitioners encounter in many planning and decision processes, for instance, the delivery and collection of containers/cargos etc.

Objective function is:

\[
\text{Minimize } Z = \sum_{k=1}^{K} \sum_{(i,j) \in A} C_{ij} x_{kij},
\]

subject to:

\[
\sum_{i=1}^{n} y_{ij} = 1, \quad j = 2,3,\ldots,n
\]

\[
\sum_{j=1}^{n} y_{ij} = 1, \quad i = 2,3,\ldots,n
\]

\[
\sum_{j=1}^{n} y_{ij} = K
\]

\[
\sum_{j=1}^{n} x_{ij} = K
\]

\[
\sum_{i=1}^{n} \sum_{j=1}^{n} D_{ij} x_{kij} \leq U, \quad k = 1,2,\ldots,K
\]

\[
\sum_{k=1}^{K} x_{kij} = y_{ij}, \quad \forall i,j
\]

\[
\sum_{(i,j) \in S \times S} y_{ij} \leq |S| - 1, \quad \text{for all subsets } S \text{ of } \{2,3,\ldots,n\}
\]

\[
x_{kij} = 0 \text{ or } 1 \quad \forall (i,j) \in A \text{ and } \forall k
\]

\[
y_{ij} = 0 \text{ or } 1 \quad \forall (i,j) \in A.
\]

A fleet of \(M\) capacitated vehicles located in a depot \((i = 1)\).

A set of target zones (of size \(N-1\)), each having demand \(D_{ij}\) \((j = 2,\ldots,N)\).

The cost \(C_{ij}\) of travelling from location \(i\) to location \(j\).

The problem is to find a set of routes for delivering/picking up goods to/from the target zones at minimum possible cost.

The vehicle fleet is homogeneous and each vehicle has a capacity of \(U\) units and variables: \(x_{kij} = 1\) if vehicle \(k\) travels on arc \(i\) to \(j\), 0 otherwise; \(y_{ij} = 1\) if any vehicle travels on arc \((i,j)\), 0 otherwise.

For further studies and fundamental features of the applied framework refer to Bish et al. (2001); Kim and Bae (1998); Vis and De Koster (2003).

- **Sheltering, storage, warehousing operations – layout design:** In a warehouse, the operating staff must decide where to locate different items of goods they receive and later deliver the items of goods. The layout problem is to determine the zones for storing each of \(n\) items that will minimize the total transportation cost between the items and the dock. For the convenience of mod-

elling, the warehouse/storage/sheltering floor area is divided into \(m\) square grids of equal size numbered from 1 to \(m\). Each grid-square can accommodate only one pallet (Sarker and Newton 2007).

Objective function is:

\[
\text{Minimize } Z = \sum_{i=1}^{n} \sum_{j=1}^{n} C_{ij} x_{ij},
\]

subject to:

\[
\sum_{j=1}^{n} x_{ij} = G_i \quad \forall i
\]

\[
\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j
\]

\[
x_{ij} \in \{0,1\} \quad \forall i,j
\]

where parameters are: \(G_i\) – the total number of grid-squares required to store item \(i\) (as an item may require more than one grid-square); \(F_i\) – the average number of pallet loads for item \(i\) received and delivered in a year; \(D_{ij}\) – the distance between the dock and the centre of grid-square \(j\); \(P_{ij}\) – the cost per pallet per unit distance incurred in transporting item \(i\) between the dock and its storage region, and decision variables: \(x_{ij} = 1\) if item \(i\) is stored in grid-square \(j\), 0 otherwise.

For more detailed studies and fundamental features of the applied framework see Kim and Kim (1998); Taleb-Ibrabimi et al. (1993), Zhang et al. (2003).

- **Intermodal connections and scheduling** is the general problem of timetabling and scheduling operation planning of the intermodal area where changes in the frequent mode occur.

Objective function is:

\[
\text{Maximize } Z = \sum_{i} \sum_{j} C_{ij} x_{ij},
\]

subject to:

\[
\sum_{j \in J} x_{ij} = S_i \quad \forall i \in I
\]

\[
\sum_{i \in I} x_{ij} \leq A_{ij} \quad \forall j \in J, \quad \forall l \in L
\]

\[
\sum_{i \in I} x_{ij} \leq 1 \quad \forall j \in J, \quad \forall m \in M
\]

\[
x_{ij} \in \{0,1\} \quad \forall i \in I, \quad j \in J
\]

where parameters are: \(I\) – a set of all intermodal groups (index \(i\)); \(J\) – a set of time groups (index \(j\)); \(L\) – a set of station groups (index \(l\)); \(M\) – a set of intermodal groups in conflict (index \(m\)); \(R_l\) – a subset of intermodal groups that can be allocated to station group \(l\); \(T_m\) – a subset of intermodal groups in conflict; the \(m\)-th row of the conflict matrix; \(A_{ij}\) – the number of stations of type \(l\); \(S_i\) – the number of materials/cargoes/container in intermodal group \(i\); \(C_{ij}\) – the desirability coefficient.
of assigning intermodal groups $i$ to time groups $j$, and with decision variables $x_{ij} = 1$ if intermodal group $i$ is assigned to time group $j$, 0 otherwise.

For more dedicated studies and fundamental features of the applied framework see Gambardella et al. (2001); Stahlbock and Voß (2008); Steenken et al. (2004).

4. Solving an Optimization Problem Applying the CE Method: Methodology and Findings

Among operational problems, for testing the purposes of the algorithm, the shortest tour problem at one short time fraction with intense terminal traffic conditions and dynamically assigned tour node scenario has been considered. Multiple vessels are serviced at the terminal. Quay cranes charge and/or discharge containers in the berthing and marshalling areas. A typical loading and unloading operation of containers at seaport terminals involves quay cranes in charging or discharging operation, multiple-trailers (ASCs, AGVs, ALVs, etc.) with loaded/unloaded containers and stacking/gantry cranes at the yard area for delivering containers to/from the stacking area (see Figs 1 and 2).

The dynamically managed seaport seaside and yard side operations involve multiple-trailers picking up containers from quay cranes (QCs) during discharging operations. Then, trailers (ASCs, AGVs, ALVs, etc.) deliver containers to the yard area and the assigned stack area for discharged containers. After delivery operations, the trailers can visit another discharging quay crane and/or trailers can visit the assigned stack area in the yard for export containers. As such, dynamic routes increase the productivity of terminal services.

The problem of the shortest tour at one short time fraction with intense terminal traffic conditions and thus dynamically assigned tour nodes for dynamic yard operations (nodes network) can be modelled by graph $G = (V, E)$ comprising a set of vertices or nodes $V$ and a set of $E$ of edges or lines. A tour to the yard area within dynamically assigned tour nodes can be represented via

![Diagram of seaport terminal operations: the seaside and yard area](image1)

**Fig. 1.** Seaport terminal operations: the seaside and yard area

![Diagram of seaport terminal operations: the yard and land area](image2)

**Fig. 2.** Seaport terminal operations: the yard and land area
permutation \( \tau = (\tau_1, \tau_2, \ldots, \tau_n) \). The shortest tour to the yard area is formulated as:

Objective function is:

\[
\text{Minimize } Z = \sum_{(i,j) \in A} C_{ij} x_{ij},
\]

subject to:

\[
\sum_{(j,i) \in A} x_{ji} - \sum_{(i,j) \in A} x_{ij} = -1
\]

if \( i = s \), 0 if \( i \neq s \) or \( d \)

\[ \forall i \in N, \ 1 \text{ if } i = d \]

\[ x_{ij} \geq 0 \quad \forall (i,j) \in A, \]

where: \( N \) – a set of the number of nodes at a seaport terminal (seaside nodes and yard area/stacking area nodes); \( A \) – a set of the existing arcs \((i,j)\); \( C_{ij} \) – arc length (or arc cost) united with each arc \((i,j)\); \( i = s \) for source node or \( i = d \) for destination node; \( x_{ij} \) is the flow from node \( i \) to node \( j \).

The objective function is to minimize the total distance dynamically defined on the seaside and yard area (Fig. 3).

For example, to solve the optimization problem, 22 nodes \((x, y)\) pairs are chosen randomly applying to the Cartesian coordinate system \((xy)\) plane (see Fig. 4) where \((x, y)\) pairs two dimensionally represents the seaside and yard area charging/discharging locations of the seaport terminal. \( Y \) axis in Fig. 4 represents the berthing area and nodes on \( y \)-axis are the location of cranes with charging/discharging containers.

\[
X = \begin{bmatrix}
5.0, 1.5, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 4.0, 7.0, 0.0, 1.0, 6.0, 4.0, 2.0, 0.0, 4.0, 3.3, 9.2, 8.0, 7.1, 6.5
\end{bmatrix}
\]

\[
Y = \begin{bmatrix}
3.0, 1.5, 2.0, 3.0, 4.0, 5.0, 7.0, 9.0, 5.0, 6.0, 4.0, 3.0, 7.0, 4.0, 9.0, 10.0, 7.7, 4.3, 6.7, 3.4, 9.9, 8.0
\end{bmatrix}
\]

In Euclidean system, if two points are \( p = (p_1, p_2) \) and \( q = (q_1, q_2) \), then the distance \( d(p, q) \) between \( p \) and \( q \) is:

\[
d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}.
\]
tances between two nodes and lighter cells depict nearby
distances between two nodes.
First, generating the initial transition matrix $p^0$ for
22 sample nodes in the form is shown below:

$$p^0 = \begin{bmatrix}
0 & p_{(1,2)} & \cdots & p_{(1,22)} \\
p_{(2,1)} & 0 & \cdots & p_{(2,22)} \\
\vdots & \vdots & \ddots & \vdots \\
p_{(21,1)} & p_{(21,2)} & 0 & p_{(21,22)} \\
p_{(22,1)} & p_{(22,2)} & \cdots & 0
\end{bmatrix}$$

where: $p^0$ has zeros in the diagonal and all remaining
elements are equal as calculated by $1/(N-1)$ which is
$1/(22-1) = 0.0476$. The rows and columns of the matrix
($22 \times 22$ matrix) add up to 1. Thus, any route at first has
equal likelihood (see Fig. 7) to be generated:

$$p^0 = \begin{bmatrix}
0 & 0.0476_{(1,2)} & \cdots & 0.0476_{(1,22)} \\
0.0476_{(2,1)} & 0 & \cdots & 0.0476_{(2,22)} \\
\vdots & \vdots & \ddots & \vdots \\
0.0476_{(21,1)} & 0.0476_{(21,2)} & 0 & 0.0476_{(21,22)} \\
0.0476_{(22,1)} & 0.0476_{(22,2)} & \cdots & 0
\end{bmatrix}$$

As in the dynamics of the CE algorithm, at each it-
eration, better $p$ vectors ($p^0, p^1, \ldots$) are created and each
of these vectors are used to generate better $z$ ($z^0, z^1, \ldots$)
values. Table shows iterations from 1 to 33. The algo-
rithm stops when $z$ converges to the global optimum
value $z^*$. Thus, the total distance reduces gradually. An op-
timal solution was found in the last (i.e. 33rd) iteration
(see Fig. 8 and Table) and an optimal tour was obtained
at the end of the last iteration.

Fig. 6. The initial parameters used for testing the CE method solving the problem of the shortest tour (at MatLab®)

Fig. 7. The initial probability transition matrix $p^0$ generated for 22 chosen sample nodes ($22 \times 22$)
Fig. 9 depicts the dynamics of the CE algorithm at each of the iterations as the sequence of matrices for the problem of the shortest tour along with the generated matrices of $P_0, P_1, P_2, \ldots$ where, during the last iteration, an optimal tour (minimum distance) has been reached.

5. About the CE Method

A comparison of algorithms from different theoretical and empirical categories is a complicated task owing to the fact that it is not a specific empirical baseline that enables an unbiased comparison among algorithms. For instance, the CE method and the genetic algorithm (GA) method are the same population-based heuristic methods. The CE uses an effective learning method throughout the search, whereas the GA method enhances the created samples from generation to generation. The GA method uses genetic encoding which is suited to a particular use of some problems and its processing time is much longer while solving small-scale problems.

Other metaheuristic algorithms (Aarts and Korst 1989; Goldberg 1989; Dorigo et al. 1999; Ehrgott and Gandibleux 2002) such as simulated annealing (SA), tabu search (TS), ant colony optimization (ACO), particle swarm optimization (PSO), memetic algorithms (MA) etc. are quite common to solving numerous kinds of problems. On the other hand, they reflect important differences that originate from theoretical and empirical grounds of algorithms. For instance, the main distinction between the CE method and SA is that the latter can be considered as a local search algorithm while on the contrary, CE is a global search one. This means that the CE method continuously seeks the global optimal solution across the big picture; on the other side, the SA method may fail to provide the global optimal and be unable to progress with the task by trapping to the state or condition of a local optimal solution.

Table. The best solution history obtained applying the CE method

<table>
<thead>
<tr>
<th>Iteration Number</th>
<th>Total Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>72.51</td>
</tr>
<tr>
<td>2</td>
<td>63.94</td>
</tr>
<tr>
<td>3</td>
<td>60.39</td>
</tr>
<tr>
<td>4</td>
<td>57.43</td>
</tr>
<tr>
<td>5</td>
<td>55.77</td>
</tr>
<tr>
<td>6</td>
<td>54.66</td>
</tr>
<tr>
<td>7</td>
<td>52.51</td>
</tr>
<tr>
<td>8</td>
<td>52.51</td>
</tr>
<tr>
<td>9</td>
<td>52.51</td>
</tr>
<tr>
<td>10</td>
<td>51.23</td>
</tr>
<tr>
<td>11</td>
<td>51.23</td>
</tr>
<tr>
<td>12</td>
<td>46.24</td>
</tr>
<tr>
<td>13</td>
<td>46.24</td>
</tr>
<tr>
<td>14</td>
<td>45.03</td>
</tr>
<tr>
<td>15</td>
<td>45.03</td>
</tr>
<tr>
<td>16</td>
<td>44.88</td>
</tr>
<tr>
<td>17</td>
<td>43.26</td>
</tr>
<tr>
<td>18</td>
<td>40.26</td>
</tr>
<tr>
<td>19</td>
<td>40.26</td>
</tr>
<tr>
<td>20</td>
<td>40.26</td>
</tr>
<tr>
<td>21</td>
<td>39.87</td>
</tr>
<tr>
<td>22</td>
<td>39.87</td>
</tr>
<tr>
<td>23</td>
<td>39.87</td>
</tr>
<tr>
<td>24</td>
<td>39.53</td>
</tr>
<tr>
<td>25</td>
<td>39.11</td>
</tr>
<tr>
<td>26</td>
<td>39.04</td>
</tr>
<tr>
<td>27</td>
<td>39.04</td>
</tr>
<tr>
<td>28</td>
<td>39.04</td>
</tr>
<tr>
<td>29</td>
<td>39.04</td>
</tr>
<tr>
<td>30</td>
<td>39.04</td>
</tr>
<tr>
<td>31</td>
<td>39.04</td>
</tr>
<tr>
<td>32</td>
<td>39.04</td>
</tr>
<tr>
<td>33</td>
<td>39.04</td>
</tr>
</tbody>
</table>

Fig. 8. The best solution history obtained applying the CE method
6. Conclusions

A wide variety of technical terminal equipment, layouts, facilities, resources and proficient workers exist at seaport terminals. The numbers of resources utilized at seaport terminals add a multitude of complexities to dynamic optimization problems since in such dynamic environments, there has been a need for solving complex operational problems within a short computing time in order to increase terminal service efficiency and to achieve improved competitiveness.

Based on the number and complexity of seaport processes, obtaining optimal solutions employing heuristic methods is a non-deterministic polynomial-time (NP) hard problem and computational time exponentially increases depending on the number of resources involved in dealing with the problem. It can be particularly clearly stated that the CE algorithm approach provides stable solutions to discovering optimal values. By utilizing and running the proposed high performing CE algorithm for the problems of seaport terminals, it seems...
to be apparent there will be significant improvements in seaport terminal services.

Acknowledgement

The authors kindly acknowledge The Graduate School of Natural and Applied Sciences, Marine Sciences and Technology (Fen Bilimleri Enstitusu, Deniz Bilimleri ve Teknoloji), at Dokuz Eylul University for the support of this study.

References


Liu, J.; Wan, Y.-W.; Wang, L. 2006. Quay crane scheduling at container terminals to minimize the maximum relative tardiness of vessel departures, Naval Research Logistics (NRL) 53(1): 60–74. doi:10.1002/nav.20108


Lokuge, P.; Alahakoon, D. 2007. Improving the adaptability in automated vessel scheduling in container ports using in-


