Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 14 (24): 3446-3454, 2014
ISSN 1812-5654 / DOI: 10.3923/jas.2014.3446.3454

Reduction Operators for Magnetic Optimization Algorithm

Mohammad Mobasher Kashani and Masri Ayob
Data Mining and Optimization Research Group (DMO), Centre for Artificial Intelligent Technology (CAIT), Universiti Kebangsaan Malaysia (UKM), Bangi Selangor, 43600, Malaysia

Abstract: Solving Berth Allocation Problem (BAP) plays an important role at the container terminals due to the increase demands in global supply chains. Since the problem is a NP-Hard problem, manual scheduling is inapplicable in the real world applications. Thus, computer based scheduling have become more popular and several approaches such as metaheuristic have been used for solving berth allocation problem. However, each algorithm has its own weakness in solving BAP such as premature convergence in particle swarm optimization and unbalance between exploration and exploitation in genetic algorithm. Based on literature, Magnetic Optimization Algorithm (MOA) has presented a better searching approach, compared to Genetic Algorithm (GA) and Particle Swarm Optimization (PSO). Therefore, this study aims to investigate the effectiveness of applying MOA in solving BAP. In addition, two reduction operators are introduced for improving MOA search process. The performance of MOA is evaluated on a real data extracted from literature (from Gioia Tauro in Italy) and compared it with GA. Results demonstrate that the MOA performances are better than GA and pseudo-simulated annealing operator can be considered as a potential component for enhancing MOA viability. This indicates the applicability of MOA in solving BAP.

Key words: Magnetic optimization algorithm, simulated annealing operator, berth allocation problem, combinatorial problems

INTRODUCTION

Maritime container terminals are principal parts of global supply chains. Terminal operator encounters several challenges on daily operations to remain competitiveness and efficiency. The majority of these challenges stem from the way different facilities and operations interact in the terminals. In order to stay competitive and avoid shipment delay, berth planning should be developed accurately and reliably. Berth planning refers to the allocating vessels in the maritime terminal to the appropriate physical locations for the process of loading and uploading in the port.

Genetic algorithms were applied on BAP by Imai *et al.* (2001) and Nishimura *et al.* (2001). In Imai *et al.* (2007) model, every berth can serve at most two vessel. By integer linear formulation, they solved problem with genetic algorithm. However, Elbeltagi *et al.* (2005) proved that Genetic Algorithms yield poor result in terms of solution quality compared to Particle Swarm Optimization method.

Tayarani and Akbarzadeh-T (2008) proposed Magnetic Optimization Algorithm (MOA) which is inspired by magnetic field theory in physics and deal with attraction between particles. Torshizi and Tayarani-N. (2010) proposed a new initialization method for MOA and Mirjalili and Sadiq (2011) implemented MOA on artificial neural network for training weights. So far, magnetic optimization algorithm has been implemented on 14 numerical function problems and artificial neural network, however, no study has been conducted on applying MOA on combinatorial problems. Thus, for discovering weakness and strength of Magnetic Optimization Algorithm, the performance over a combinatorial problems (e.g., BAP) need to be observed which it can encourage researchers to study the performance of MOA on combinatorial problem field.

Magnetic optimization algorithm has two constant variables which are used to balance between exploration and exploitation in the search space. These constant variables are set at the initialization phase then used during the search process. In this study, linear reduction and pseudo-SA operators are proposed for assigning two constant variables of MOA to increase exploration at early iterations and exploitation toward end. Basic MOA with two proposed reduction operators are applied on BAP and results are tested on dataset from Gioia Tauro port of Italy by Cordeau *et al.* (2005) and compared to GA which is presented by Yang *et al.* (2012).

Corresponding Author: Mohammad Mobasher Kashani, Data Mining and Optimization Research Group (DMO), Centre for Artificial Intelligent Technology (CAIT), Universiti Kebangsaan Malaysia (UKM), Bangi Selangor, 43600, Malaysia

Recently, growing trend toward Berth allocation problem has been seen and it puts pressure on global supply chain and it shows the importance of scheduling vessels on seaside studies. By having a better schedule, one can reduce the weighing, total handling time and/or can have an optimized location allocation in different quay. Thus, we are given a set of vessels with handling time and certain (or uncertain) arrival time. Our mission is to assign every vessel to an appropriate berth in the quay layout for obtaining an optimum scheduling by which the cost of transshipment reduces. So far, a big number of models have been introduced and several methods on different models have been proposed.

In order to schedule vessels in a quay we need some basic informations from berths and vessels. These data vary greatly depending on method, formulations and objectives. The objectives range from minimization of vessel waiting time, total service times, rejected vessel by some constraints to minimization of difference between planned and actual berthing schedules. There are a great number of temporal and spatial additional constraints involved in the Berth Allocation Problem, that several BAP formulations derive from these constraints. The current models for Berth allocation problem in literature can be categorized based on the spatial aspects like draft restriction, berth layout as well as temporal aspects like handling and service starting time and arrival process of vessel. Berth allocation problem has been proved to be an NP-hard problem by changing to the single machine scheduling problem by Hansen and Oguz (2003), set partitioning problem by Lim (1998) and cutting stock problem in two dimension by Imai *et al.* (2005).

Spatial constraints limit the vessel berthing positions according to a pre-set berth partitioning. The following different cases are discussed by Imai *et al.* (2005).

- The discrete quay is divided into a number of partitions, called berths. Every berth has the capacity of maximum one vessel. Berth location can be determined by the quay layout (Fig. 1a) or in order to planning simplicity, can be organizationally prescribed (Fig. 1b)
- In the continuous scheme quay is not divided into certain sections, i.e., as long as vessels don't violate the boundaries of quay, vessels can be served in any arbitrary places (Fig. 1c). Although continuous scheme uses quay space more economically, discrete scheme is simpler in terms of implementation
- In hybrid scheme, due to difference between sizes of vessels, large vessel may be bigger to fit in a berth (Fig. 1d) or small vessels can share berths (Fig. 1e). In hybrid scheme a berth can be shared by small berths, large vessels use two berths or in indented berth can be served from both side (Fig. 1f)

For applying draft limitation, spatial constraints like water depth relating to the vessel type must be taken into account. To prevent exceeding the handling time, vessels usually remain at the assigned berthing location during the whole service. On the contrary, in some literature vessels can relocate in quay (Brown *et al.*, 1994, 1997; Lee and Chen, 2009).

In order to limit the departure time and berthing time, temporal constraints are used in berth planning. The following different cases are discussed by Imai *et al.* (2001):

- **Condition 1:** Static arrival time where there are two cases in static arrival time: In the first case, there is no arrival time and in the second case, arrival time is considered as a soft constraint. The assumption for the first case is that all vessels are present at the quay and can be served immediately. On the other hand, in the second case it is assumed that, by regarding a certain cost, vessels can be served earlier than the due arrival time

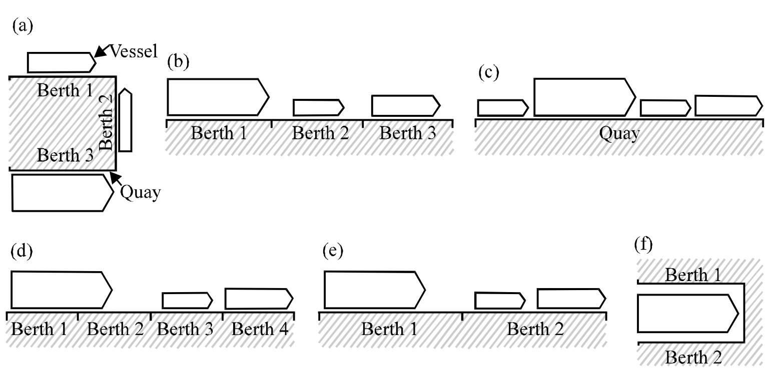

Fig. 1(a-f): Different berth layout (Imai *et al.*, 2005)

- **Condition 2:** Dynamic arrival time in which vessels have fixed arrival times and cannot be served before due arrival time

In dynamic arrival time case, the total vessel services must be completed within the time window. Due dates can be defined as an extra constraint or determined by the latest vessel departure time. In the majority of literature of BAP models, handling times of vessels are considered deterministic. Still, researchers take into account different ways of dealing with handling times:

- **Condition 3:** Handling times are fixed and known in advance
- **Condition 4:** Handling times are calculated by regarding to the vessels berthing position
- **Condition 5:** Handling times are based on the assigned cranes schedules
- **Condition 6:** In some cases also conditions 2, 3 and 4 can be combined

Solving optimization problems has been of great importance in various fields, namely construction, resource allocation and job scheduling. Researchers used to solve the problems by exhaustive methods (brute-force search) which was time consuming and in some cases impossible to solve. Consequently, numerous iterative procedures (algorithms) have been proposed by researchers to obtain a better solution in a feasible time for optimization problems. Generally, these algorithms can be categorized as problem-specific and metaheuristic methods. Problem-specific algorithms are methods which are designed specifically for a certain problem and are not be able to implement on other problem. In this sense, every new problem needs its algorithm and it makes it difficult to solve problems. On the other hands, metaheuristic algorithms claim to be able to solve various types of optimization problems in feasible time.

Optimization problems are divided into two main classes: First class is those which have continuous variables and the second one is problems with discrete variables, which is called combinatorial problems. The main difference between these two classes is that in the continuous problems we aim to find real numbers as objective values, on the other hands, in combinatorial problems we try to find a finite or countably infinite values. Due to the difference between the natures of these two problems, optimization methods take different approaches for solving problems in every class.

Evolutionary algorithm are continually applied to several optimization problems in the operation research, engineering marketing, social science fields including genetic, scheduling, structural design, material selection and so on. Evolutionary algorithms, as well as mathematical optimization problems, by inspiring from natural selection notion in the evolutionary process in the nature have been applied as an experimental framework.

Evolutionary algorithms (EAs) are considered metaheuristic optimization based on population. Every solution in the optimization problem is an individual in population and evolution process leads to obtaining better solutions during iterations. With regard to problem nature, fitness values of every individual are calculated and, inspired by biological evolutionary process, population evolve repeatedly.

The most popular Evolutionary Programming (EP), Evolutionary Algorithms are Evolutionary Strategy (ES) and Genetic Algorithm (GA). Holland presented Genetic Algorithm in 1970s. Genetic Algorithm uses crossover, mutation and selection operators in the evolution of population. Evolutionary Strategy was proposed by Schwefel and Rechenberg in 1960s (Schwefel, 1965; Rechenberg, 1971) has selection and mutation operators. Evolutionary programming and genetic programming are ES and GA based approaches to obtain mathematical function or computer program that perform tasks which define by user.

As related methods, Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) are famous among other algorithms. Particle Swarm Optimization (PSO) was presented by Kennedy and Eberhart (1995) i and Ant Colony Optimization (ACO) was proposed in 192 by Dorigo. Unlike Evolutionary Strategy and Genetic Algorithm that are inspired from genetic evolution, Particle Swarm Optimization and Ant Colony Optimization are from the behaviour of bird swarm and insects colony (ants), respectively. Hence, Particle Swarm Optimization and Ant Colony are often categorized into the swarm intelligence methods.

Magnetic Optimization Algorithm (MOA) which is proposed by Tayarani and Akbarzadeh-T (2008) is a novel evolutionary algorithm inspired by the electromagnetic law in physics. Electromagnetic is one of the four main forces in the universe and authors proved that implementing electromagnetic law in optimization field leads to a better results compare to PSO, GA on 14 numerical problems.

MATERIALS AND METHODS

By using the concept of electromagnetic in physics, Magnetic Optimization Algorithm which is proposed by Tayarani and Akbarzadeh-T (2008) was applied on 14 numerical benchmark function. MOA uses lattice like network for interacting between particles and the interactions are conducted by using electromagnetic force which is a combination of similarity between particles and difference between their solution values. A total force from neighbour particles determines the amount of move which the particle should have in the next step. Pseudo code of basic MOA is shown as follows:

MOA procedure:

```
1. Initial parameters
While Terminating criteria is not satisfied
2.  Evaluate particles in X^t and store their fitness values in magnetic field B^t
3.    Normalize B^t and evaluate mass M^t according to Eq. 1
          for each particle do
                F_ij←0
4.  Find neighbours based on network
      for each neighbour of corresponding particle
5.              Calculate forces from neighbours
      end
            for each particle
6.              Update velocity based on neighbour forces
7.              Update the particle based on its corresponding velocity
         end
     end
end
```

Variables which are used in this algorithm are listed below:

B_{ij}^t	:	Magnetic field in position j of particle i in iteration t
Min	:	Minimum value of magnetic field
Max	:	Maximum value of magnetic field
M_{ij}^t	:	The mass value of position j of particle i in iteration t
x_{wk}	:	Particle x from column k and row w
N_{wk}	:	Neighbour particles set from column k and row w
$D(x_{ij}^t, x_{uv}^t)$	:	Euclidian distance between particles x_{ij}^t and x_{uv}^t
v_{ij}^t	:	Velocity value of position j of particle i in iteration t
F_{ij}^t	:	Force value on position j of particle i in iteration t

MOA procedure consists of 7 main steps by Tayarani and Akbarzadeh-T (2008) which are described as follows:

- **Initial parameters:** First step is initialization of solutions for t = 0. In this study, two mode of solution initialization are discussed; randomly and First Come First Serve (FCFS) methods. Also constant parameters ρ and α are initialled in this step
- **Particle evaluation:** Fitness value in every particle is calculated in this phase then stored in magnetic field B_{ij}
- **Normalization:** In this phase, first B^t for all particles are normalized by Eq. 1 then masses are calculated according to Eq. 2:

$$B_{ij} = \frac{B_{ij} - Min}{Max - Min} \quad (1)$$

where, Min is the minimum value of B_{ij} for all particles and Max is the maximum value of B_{ij} among all particles.

Mass is calculated based on proposed equation by Tayarani and Akbarzadeh-T (2008):

$$M_{ij}^t = \rho \times B_{ij}^t + \alpha \quad (2)$$

Two constant values of ρ and α control the movement of particles for having a better balance between exploration and exploitation in different problems.

- **Finding neighbours:** In order to find velocity for every particle in the entire population, accumulation of forces from other particle to a certain agent should be estimated. First step toward calculating forces is to find particle neighbours. Tayarani and Akbarzadeh-T (2008) proposed a lattice like network in which every particle is neighbour with four other particles in the network (Fig. 2)

Neighbours of every particle are found based on connection network. For lattice-like network, neighbours are obtained as follows:

$$N_{wk} = \{x_{w'k}, x_{wk'}, x_{w''k}, x_{wk''}\} \quad (3)$$

Where:

$$w' = \begin{cases} w - 1 w \neq 1 \\ Sk = 1 \end{cases} \quad (4)$$

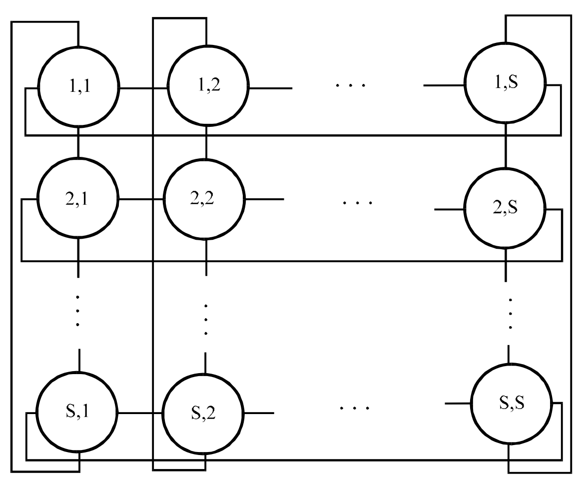

Fig. 2: Lattice like network for finding neighbour particles

$$w'' = \begin{cases} w+1 & w \neq S \\ 1 & w = S \end{cases} \tag{5}$$

$$k' = \begin{cases} k-1 & k \neq 1 \\ S & k = 1 \end{cases} \tag{6}$$

$$k'' = \begin{cases} k+1 & k \neq S \\ 1 & k = S \end{cases} \tag{7}$$

Another possible connection network for particle is fully-connected network which has never been studied before. The performance of this network in terms of running time and solution quality will be compared to lattice-network.

- **Calculating force:** The force value F_{ij} is calculated by using Eq. 8. This formulation is the combination of similarity between two solutions and magnetic field which represents fitness value of particle:

$$F_{ij} = F_{ij} + \frac{(x^t_{uv} - x^t_{ij}) \times B^t_{uv}}{D(x^t_{ij}, x^t_{uv})} \tag{8}$$

The first section of Eq. 8 is the impact of other neighbours on particle. The middle part of Eq. 8 is related to similarity of two solutions and, finally, the last section represents magnetic field. For finding geometry distance of particles, D(.,.) is determined by l^2-norm of two solutions which is shown in Eq. 9:

$$D(x^t_{ij}, x^t_{uv}) = \sqrt{\sum_{k=1}^{n} (x^t_{ij} - x^t_{uv})} \tag{9}$$

We can see from Eq. 8 that force is related to distance of two particles and fitness value of the one that we want to calculate for fitness.

- **Updating velocity:** Particles in PSO move, with regard to their local best solutions and global best, to find a better solution. Particles in MOA, similar to PSO, tend to move in every iteration and it can be achieved by using velocity which counts an amount of movement for every particle by regarding their forces and Masses. Equation 10 calculates velocity which is based on mass, accumulated forces and a random value:

$$v^{t+1}_{ij,k} = \frac{F_{ij}}{M_{ij}} \times R() \tag{10}$$

- **Updating particles:** In this phase, particle position is updated with regard to its velocity value:

$$x^{t+1}_{ij} = x^t_{ij} + vx^{t+1}_{ij} \tag{11}$$

Representation: Representation of solution is one the most fundamental parts of evolutionary algorithm. Ting *et al.* (2014) presented a representation for Particle Swarm Optimization algorithm applied on BAP. Figure 3 is an example of this representation. Solution in this case is created for 9 vessels and 3 berths. Each solution dimension is encoded as a real number between (1,4). Integer part determines the number of berth, thus, dimensions with the same integers belong to the same berth. The fractional part of real number represents the order of vessel in the berth. For example, in Fig. 4 vessels {7, 8, 9} belongs to berth {1}. After sorting their values the right order of vessels changes to {7, 9, 8}. It means vessel 7 serves first then vessel number 9 and finally 8.

BAP Formulation: Based on BAP models by Cordeau *et al.* (2005), in this study, dynamic and discrete form of BAP modelled as Multi-Depot Vehicle Routing Problem with Time Window (MDVRPTW). MDVRPTW deals with the problem of delivering uniform goods to customers from a set of depots with certain capacity. The vessels are considered as customers and berths are seen as depots in which one vehicle is placed. There are m vehicles for m depots (one vehicle for each depot) and every vehicle begins and ends its tour at its depot. From geometry point of view, in two dimensions graph, vessels are presented with vertices based on their length and depots are modelled as origin and destination vertices with regard to their availabilities on time windows. The multi-graph model is given with $G^k = (V^k, A^k)$, $\forall k \in M$ where, $A^k \subseteq V^k \times V^k$ and $V^k = N \cup \{o(k), d(k)\}$. Input variables are given by:

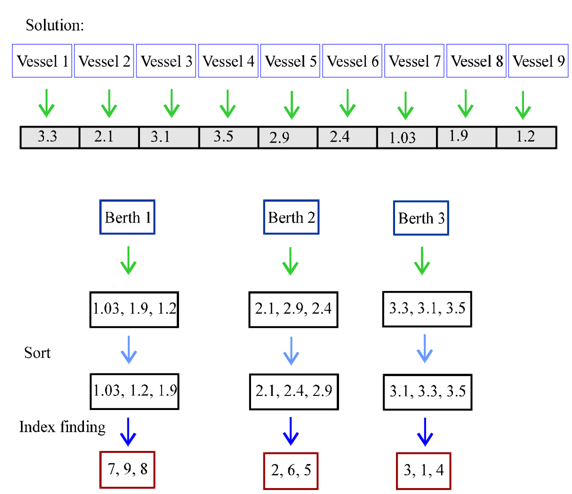

Fig. 3: Representation of particles in MOA

N : Set of vessels, n = |N|
M : Set of berths, m = |M|
t_j^k : Handling time of vessel j at berth k
a_j : Arrival time of vessel j
s^k : Start of availability time for berth k
e^k : Finish of availability time for berth k
b_j : Upper bound for service time window for berth j

The model variables are defined as:

- $x_{ij}^k \in \{0,1\}$, $(i,j)\in A^k$, $k\in M$; $x_{ij}^k = 1$ when vessel j scheduled after vessel i at berth k
- $T_j^k, j\in N, k\in M$: Berthing time of vessel j at berth k
- $T_{o(k)}^k, k\in M$: Starting service time of berth k (the time when the first vessel moors)
- $T_{o(k)}^k, k\in M$: Ending service time of berth k (the time when the last vessel leaves the berth)
- M_{ij}^k: $\max\{b_i - a_j + t_j^k, 0\}$, i and j∈N, k∈M

The objective function is defined as a minimization of total service times of vessels which is shown in Eq. 12 and is calculated based on total time from arrival a vessel to completing its loading/unloading on berth which is explained with more detail in Fig. 4.

Minimize:

$$\sum_{i\in N}\sum_{k\in M} v_i\left[T_i^k - a_i + t_i^k \sum_{j\in N\cup\{d(k)\}} x_{ij}^k\right] \quad (12)$$

Subject to:

$$\sum_{k\in M}\sum_{j\in N\cup\{d(k)\}} x_{ij}^k = 1 \forall i\in N \quad (13)$$

$$\sum_{j\in N\cup\{d(k)\}} x_{o(k)j}^k = 1 \forall k\in M \quad (14)$$

$$\sum_{i\in N\cup\{o(k)\}} x_{id(k)}^k = 1 \forall k\in M \quad (15)$$

$$\sum_{j\in N\cup\{d(k)\}} x_{ij}^k - \sum_{j\in N\cup\{o(k)\}} x_{ji}^k = 1 \forall k\in M \forall i\in N \quad (16)$$

$$T_i^k + t_i^k - T_j^k \le (1 - x_{ij}^k) M_{ij}^k \forall k\in M \forall (i,j)\in A^k \quad (17)$$

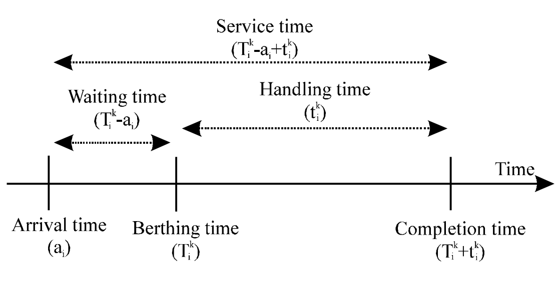

Fig. 4: Total service time of a sample vessel (De Oliveira *et al*., 2012)

$$T_i^k \ge a_i \forall k\in M \forall i\in N \quad (18)$$

$$T_i^k + t_i^k \sum_{j\in N\cup\{d(k)\}} x_{ij}^k \le b_i \forall k\notin M \forall i\in N \quad (19)$$

$$T_{o(k)}^k \ge sk \forall k\in M \quad (20)$$

$$T_{d(k)}^k \le e^k \forall k\in M \quad (21)$$

$$x_{ij}^k \in \{0,1\} \forall k\in M \forall (i,j)\in A^k \quad (22)$$

Constraint (Eq. 13) ensures that every vessel is assigned only once to once to a berth. Constraints (Eq. 14, 15) state that for every berth k a vessel will be the first and other will be the last to be moored. Constraint (Eq. 16) indicates flow conservation the rest of vessels and constraint (Eq. 17) defines the consistency of berthing time for vessels. Since each vessel can be served on only certain berths (vessel size of type of containers may not be appropriate for vessel to be moored on some berths), just valid vertices A^k, ∀ k∈M are allowed in constraint (Eq. 17). Constraints (Eq. 18, 19) guarantee that berthing time will be after arrival time a vessel and completion time will be before vessel time window. Constraints (Eq. 20, 21) avoid violation in berths by controlling time window and constraint (Eq. 22) ensures that the decision variable x_{ij}^k will be binary.

Improved MOA: Two constant values of ρ and α control the movement of particles for having a better balance between exploration and exploitation. The higher value of ρ and α results is the slower movement, due to heavy mass, which causes more exploitation and the less ρ and α causes faster movement, which causes more exploration. In this study two operator are tested to assign two constant values of ρ and α automatically during search process. The main idea behind this assignment is to provide a method which can increases exploration and decreases exploitation at early generations then declines exploration and rises exploitation gradually in the following iterations. This kind of parameter assignment will cause more effective search that can generate good quality solution. First operator for this reason is Linear Reduction which is presented in Eq. 23:

$$M_{ij}^t = \frac{B_{ij}^t}{W_{max} - \left(\frac{W_{max} - W_{min}}{Max_t} \times t\right)} \quad (23)$$

where, t is current iteration, Max_t is the maximum iteration and W_{max} and W_{min} are two constant for adjusting this operator for different problems. This operator reduces exploration and increases exploitation linearly through generations.

Since the nature of some problems are not linear, pseudod-SA operator is used in this study as another substitution for two constants ρ and α to have more flexible approach for different situations. SA method which is used in this research is as follows:

$$M_{ij}^{t} = \frac{B_{ij}^{t}}{M' \times e^{\left(\frac{t \times \alpha'}{Max_t}\right)}} \tag{24}$$

where, Max_t is temperature which is maximum iteration in MOA, M' is a constant coefition and α′ is cooling rate which are used for adjusting the operator in different search spaces. Equation 24 is inspired by SA formula (Kirkpatrick *et al*., 1983) by adding constant value M' to obtain more flexibility in different situations.

RESULTS AND DISCUSSION

The first step toward analyzing MOA is to find the best parameter values by tuning through different tests. Tayarani and Akbarzadeh-T (2008) proposed two parameters, ρ and α, for balancing between exploration and exploitation in different problems, hence, these parameters need to be tuned for BAP. For more clear presentation of tuning results of basic MOA, Fig. 5 is designed to demonstrate the fitness value correspond to its ρ and α values in three dimensional space. As shown in Fig. 5, x axis is ρ values, y axis represents α values and z axis is fitness values for each ρ and α. According to this figure, it is clear that for ρ = 10 and α = 15, basic MOA yields best result.

In this part two methods for calculating ρ and α coefitions are tested and compared to the constant ones. In the first place, parameters of SA and LR should be tuned. We start the tuning by SA and test it by following default parameters. Note that unlike agents (Tayarani and Akbarzadeh-T, 2008) are fully connected to each other. Table 1 shows the default parameters for running SA on MOA.

Based on the Eq. 4 there are two parameters which should find their optimal values. To achieve this we designed the experimental as follow:

Figure 6 illustrates the 3D presentation of parameter tuning for pseudo-SA in basic MOA. In this plot, x axis is M and y axis presents α and finally z axis is corresponding fitness of every x and y. Based on this plot we can see that test with M = 1 and α = 5 yields the best result in compare to other cases.

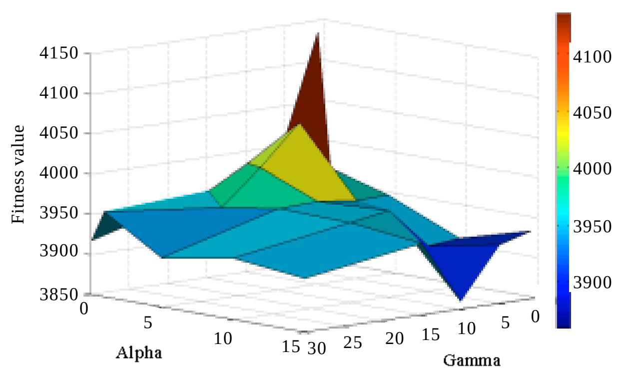

Fig. 5: 3D presentation of ρ and α tuning

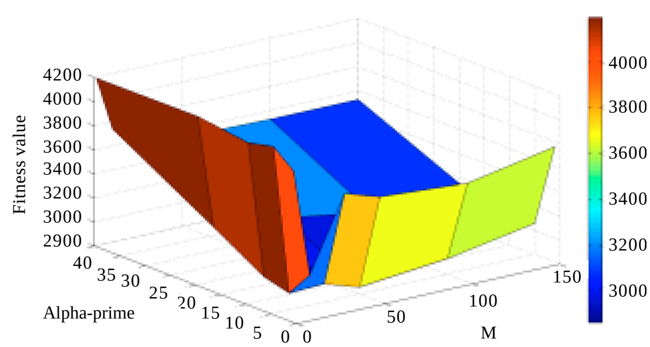

Fig. 6: 3D presentation of M and α' tuning for MOA with SA

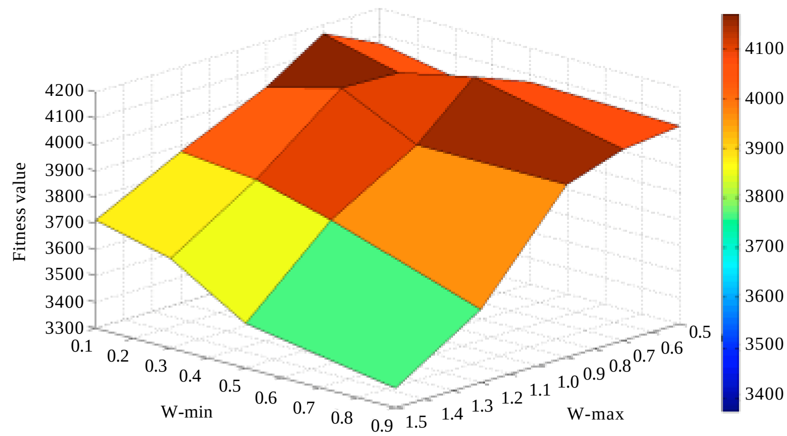

Fig. 7: 3D presentation of LR approach parameters tuning

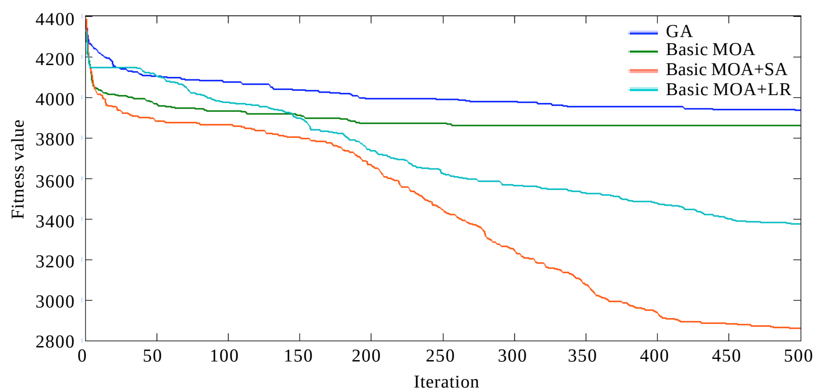

Fig. 8: Comparison between MOA approaches with GA

Table 1: Setting default parameters for running SA test

Model	Run	Iteration	Population	Dataset
Pseudo-SA	31	500	30	I3/i01

Table 2: Setting default parameters for MOA with LR

Model	Run	Iteration	Agent	Dataset
Linear reduction	31	500	30	I3/i01

Table 3: Comparison between different MOA approaches with GA

Method	Fitness	STD	Time (sec)
GA	3934.3	121.69	5.2126
Basic MOA	3859.5	186.16	4.2404
Basic MOA+LR	3372.8	343.61	4.4566
Basic MOA+SA	2857.4	230.39	4.2158

In the next phase of tuning process, Linear Reduction method is considered as a substitute for ρ and α in MOA formula. Algorithm parameters are set in Table 2. Agents are fully connected in this experiment.

According to Eq. 1, two LR parameters should be tuned during this step (W_{max} and W_{min}). Parameters for test are designed as follow:

Figure 7 illustrates the 3D presentation of parameter tuning for LR in basic MOA. In this plot, x-axis is W_{max} and y axis presents W_{min} and finally z axis is its corresponding fitness value of every x and y.

As can be seen in Fig. 7 for W_{max} 0.7 and W_{min} 0.9 the method gives its best result in compare to other test values. Results are obtained from 20 different tests and for every test, mean fitness value, standard deviation and mean running time are calculated.

From the data in the Table 3 and by comparing to LR approach, it is clear that LR is a better method for calculating ρ and α during MOA search process than basic MOA but worse than basic MOA with SA. By comparing the role of SA and LR as a substitution for two constants ρ and α in the basic MOA, we can conclude that SA is a better approach for calculating ρ and α.

To have a better insight into exploration and exploitation of results in Table 3. Figure 8 provides a comparison between mentioned methods with same maximum iteration and population size. It is clear that pseudo-SA operator provides a better exploration/exploitation during the search process and solution quality is improved considerably in compare to basic MOA with LR operator and GA.

CONCLUSION

This study focused on the application of optimization algorithms on combinatorial problems. One of the most important objectives of this research was analyzing the performance of MOA on a NP-Hard problem (BAP). From this point of view, two new operators, pseudod-SA and Linear Reduction, for calculating the mass in the MOA was introduced and their viabilities on different situation were tested. It can be seen that basic MOA with pseudo-SA provides a better solution quality.

For future research, study on improving the performance of the proposed method in terms of running time and standard deviation will be beneficial.

ACKNOWLEDGMENT

The authors wish to thank the Universiti Kebangsaan Malaysia for supporting this work under the "Industrial and Community Research Grant Scheme" (INDUSTRI-2013-034) and Privasia Technology Bhd.

REFERENCES

Brown, G.G., K.J. Cormican, S. Lawphongpanich and D.B. Widdis, 1997. Optimizing submarine berthing with a persistence incentive. Naval Res. Logistics, 44: 301-318.

Brown, G.G., S. Lawphongpanich and K.P. Thurman, 1994. Optimizing ship berthing. Naval Res. Logistics, 41: 1-15.

Cordeau, J.F., G. Laporte, P. Legato and L. Moccia, 2005. Models and tabu search heuristics for the berth-allocation problem. Transp. Sci., 39: 526-538.

De Oliveira, R.M., G.R. Mauri and L.A. Nogueira Lorena, 2012. Clustering search for the berth allocation problem. Exp. Syst. Appl., 39: 5499-5505.

Elbeltagi, E., T. Hegazy and D. Grierson, 2005. Comparison among five evolutionary-based optimization algorithms. Adv. Eng. Inform., 19: 43-53.

Hansen, P. and C. Oguz, 2003. A note on formulations of the static and dynamic berth allocation problems. Report, Les Cahiers du Gerad, G-2003-20, HEC, Montreal, Quebec, Canada.

Imai, A., E. Nishimura and S. Papadimitriou, 2001. The dynamic berth allocation problem for a container port. Transp. Res. B: Methodol., 354: 401-417.

Imai, A., E. Nishimura, M. Hattori and S. Papadimitriou, 2007. Berth allocation at indented berths for mega-containerships. Eur. J. Operat. Res., 179: 579-593.

Imai, A., X. Sun, E. Nishimura and S. Papadimitriou, 2005. Berth allocation in a container port: Using a continuous location space approach. Transp. Res. B: Methodol., 39: 199-221.

Kennedy, J. and R. Eberhart, 1995. Particle swarm optimization. Proc. IEEE Int. Conf. Neural Networks, 4: 1942-1948.

Kirkpatrick, S., C.D. Gelatt and M.P. Vecchi, 1983. Optimization by simmulated annealing. Science, 220: 671-680.

Lee, Y. and C.Y. Chen, 2009. An optimization heuristic for the berth scheduling problem. Eur. J. Operat. Res., 196: 500-508.

Lim, A., 1998. The berth planning problem. Operat. Res. Lett., 22: 105-110.

Mirjalili, S. and A. S. Sadiq, 2011. Magnetic optimization algorithm for training multi layer perceptron. Proceedings of the IEEE 3rd International Conference on Communication Software and Networks, May 27-29, 2011, Xi'an, China, pp: 42-46.

Nishimura, E., A. Imai and S. Papadimitriou, 2001. Berth allocation planning in the public berth system by genetic algorithms. Eur. J. Operat. Res., 131: 282-292.

Rechenberg, I., 1971. Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution. Ph.D. Thesis, Department of Process Engineering, Technical University of Berlin, Germany.

Schwefel, H.P., 1965. Kybernetische evolution als strategie der exprimentellen forschung in der stromungstechnik. Master's Thesis, Technical University of Berlin, Germany.

Tayarani, N.M.H. and M.R. Akbarzadeh-T, 2008. Magnetic optimization algorithms a new synthesis. Proceedings of the IEEE World Congress on Evolutionary Computation, June 1-6, 2008, Hong Kong, pp: 2659-2664.

Ting, C.J., K.C. Wu and H. Chou, 2014. Particle swarm optimization algorithm for the berth allocation problem. Expert Syst. Applic., 41: 1543-1550.

Torshizi, M. and M. Tayarani-N., 2010. Functional sized population magnetic optimization algorithm. Proceedings of the 4th International Workshop on Natural Computing, September 23-25, 2009, Himeji, Japan, pp: 316-324.

Yang, C., X. Wang and Z. Li, 2012. An optimization approach for coupling problem of berth allocation and quay crane assignment in container terminal. Comput. Ind. Eng., 63: 243-253.