Mazandaran
University of
Science and
Technology

Available online at www.iiec2017.com

13th International Conference on Industrial Engineering
(IIEC 2017)

Iran
Institute of

Addressing the Vehicle Routing Problem in Large Scale Network via Novel Nature-inspired Algorithm

---

M. Fazli[1*], **M. Hajiaghaei-Keshteli**[2], **A. M. Fathollahi Fard**[2]

[1]*Department of Industrial Engineering, Shomal University, Amol, Iran*
[2]*Department of Industrial Engineering and Management Systems, University of Science and Technology of Mazandaran, Behshahr, Iran*

Abstract

Nowadays, nature-inspired plays a key role especially in transportation problems and operation management problems. This study presents the application of a novel nature-inspired algorithm to the basic Vehicle Routing Problem (VRP) in the large scale network, in which customers of known demand are supplied from a single depot. Vehicles are subject to a weight limit and, in some cases, to a limit on the distance travelled. Only one vehicle is allowed to supply each customer. Furthermore, to address the proposed problem, RDA inspired by Red Deer's mating and presented recently as one of novel metaheurisitc methods is introduced. This issue aims to explore the RDA on the proposed VRP for the first time. In addition, computational results are computed for the RDA which is put forward. Further results are given using an imperialist competitive algorithm and genetic algorithm, showing that this approach is competitive with other strong metahuristics in terms of solution time and quality.

Key-words: Vehicle Routing Problem (VRP), Metaheuristics, Red Deer Algorithm (RDA).

1. Introduction

The basic vehicle routing problem (VRP) consists of a number of customers, each requiring a specified weight of goods to be delivered. Vehicles dispatched from a single depot must deliver the goods required, then return to the depot. Each vehicle can carry a limited weight and may also be restricted in the total distance it can travel. Only one vehicle is allowed to visit each customer. The problem is to 2nd a set of delivery routes satisfying these requirements and giving minimal total cost. In practice, this is often taken to be equivalent to minimizing the total distance travelled, or to minimizing the number of vehicles used and then minimizing total distance for this number of vehicles [1].

Most published research for the VRP has focused on the development of heuristics. Although the development of modern heuristics has led to considerable progress, the quest for improved performance continues. In this regard, RDA as one of new metaheuristics is implemented. The Scottish Red Deer (*Cervus Elaphus Scoticus*)

---

[1] Corresponding authors:
Tel: +98 11 34552000 (263), Fax: +98 11 34552008
Masoudfazli97@yahoo.com (M. Fazli), mostafahaji@mazust.ac.ir (M. Hajiaghaei-Keshteli), amirfard@mazust.ac.ir (A. M. Fathollahi Fard).

is a subspecies of Red Deer, which lives in the British Isles [2]. In breeding season, male RDs roar loudly and repeatedly. Roaring male RDs causes to attract hinds. Females prefer a high to a low roaring rate [3]. Red Deer mating patterns usually involve a dozen or more mating attempts before the first successful one. There may be several matings before the stag (the mature male) will seek out another mate in his harem. A harem is a group of females, which mate with the head of harem (male commander). The commander occupies the territory and protects the other hinds in his harem. The more hinds in the harem, the more the power of male commander. Like other metaheuristics, the algorithm starts with an initial population which is called RDs. We divided them into two groups; male RDs and hinds. The males roar and after we choose the best males as male commanders. The commander mates with high percent of females and make a new generation. This procedure accompanies with some mating and fighting behavior make this intelligent evolutionary algorithm.

The rest of papers are summarized as follows. Section 2 explored the proposed mathematical model exactly. RDA and its steps are detailed in section 3. The performance and efficiency of RDA in the proposed model are developed in section 4. At the least, the results and future works are presented in section 5.

2. Problem description

As mentioned earlier, we consider the problem of routing vehicles stationed at a central facility (depot) to supply customers with known demands, in such a way as to minimize the total distance travelled. The problem is referred to as the vehicle routing problem (VRP) and is a generalization of the multiple travelling salesman problem that has many practical applications. The problem is happened when some customers are visited each other by some vehicles. In basic of Vehicle Routing Problem (VRP) consists of a number of customers each requiring a specified weight of goods to be delivered. Vehicles detached from a single depot must deliver the goods required, then return to the depot. Each vehicle can carry a limited weight and may also be restricted in the total distance it can travel. Only one vehicle is allowed to visit each customer. The problem is to 2nd a set of delivery routes satisfying these requirements and giving minimal total cost. In practice, this is often taken to be equivalent to minimizing the total distance travelled, or to minimizing the number of vehicles used and then minimizing total distance for this number of vehicles [4].

It is so clearly that this problem is known as an NP-*hard* problem, the computational time to obtain exact solution increases in a polynomial fashion and very quickly becomes extremely difficult and long as the size of the problem increases. The problem has been solved by using heuristics and metaheuristics. Tabu search, simulated annealing, genetic algorithm, ant colony optimization and variable neighborhood search and also bat algorithm are reported to solve this problem [4-9].

We bring below a formulation of VRP by an integer programming. This formulation is a simplification of one which is given in pervious researches [4-7].

The basic of VRP considered here as follows:

A graph G=(X, A) is defined by the set X of its vertices and the set A of its arcs. Let X'={x_i| i=1, …, N} be used for the set of *N* customers and let x_0 be the depot. X=X'∪ $\{x_0\}$.

A customer x_i has the following requirements:

(a) A quantity q_i of some product to be delivered by a vehicle,

(b) A "cost" u_i required by a vehicle to unload the quantity q_i, at x_i.

We presume that M identical vehicles each of capacity Q are stationed at depot and the total "cost" (e.g. "distance" or "time") of a vehicle route must be less than or equal to a given number T.

The number of vehicles is large enough for a feasible solution to exist.

Let ε_{ijk} =1 if vehicle *k* visits customer x_j immediately after visiting customer x_i, $\varepsilon_{ijk} = 0$ otherwise.

$$\mathit{minimize}\ \ z = \sum_{i=0}^{N}\sum_{j=0}^{N} c_{ij} \sum_{k=1}^{M} \varepsilon_{ijk} \tag{1}$$

$$\mathit{subject\ to}\ \sum_{i=0}^{N}\sum_{k=0}^{M} \varepsilon_{ijk} = 1, \qquad j = 1, \dots, N, \tag{2}$$

$$\sum_{i=0}^{N} \varepsilon_{ipk} - \sum_{j=0}^{N} \varepsilon_{pjk} = 0, \quad k = 1, \dots, M, p = 0, \dots, N, \tag{3}$$

$$\sum_{i=1}^{N} (q_i \sum_{j=0}^{N} \varepsilon_{ijk}) \le Q, \ \ k = 1, \dots, M, \tag{4}$$

$$\sum_{i=0}^{N}\sum_{j=0}^{N} c_{ij}\, \varepsilon_{ijk} + \sum_{i=1}^{N} (u_i \sum_{j=0}^{N} \varepsilon_{ijk}) \le T, \ \ k = 1, \dots, M, \tag{5}$$

$$\sum_{j=0}^{N} \varepsilon_{0jk} = 1, k = 1, \dots, M, \tag{6}$$

$$y_i - y_j + N \sum_{k=1}^{M} \varepsilon_{ijk} \le N - 1, i \ne j = 1, \dots, N \tag{7}$$

$$\varepsilon_{ijk}\ \epsilon \{0, 1\}\ \mathit{for\ all}\ i, j, k, y_i\ \mathit{arbitrary} \tag{8}$$

In the above formulation, Eq. (2) explains that a customer must be seen exactly once. Eq. (3) states that if a vehicle visits a customer, it must also depart from it. Eq. (4) and Eq. (5) are the capacity and “cost” limitations each route. Eq. (6) explains that a vehicle must be used exactly once. Eq. (7) is the subtour-elimination condition. And Eq. (8) is explained the binary variable decision.

In the following section, the proposed method to tackle this NP-*hard* problem is presented.

3. Red Deer Algorithm (RDA)

Although many methods have been developed in the recent two decades, but just only a few of them considered and discussed on the two important phases; exploration and exploitation, and their trade-off. Red Deer algorithm Fathollahi Fard and Hajiaghaei-Keshteli [10] is one of first methods in recent metaheuristics to give the opportunity to a user to make a balance between intensification and diversification. This algorithm explored the Red Deer’s characteristics in breeding season and simulated their main behaviors in this specially time of year. The Scottish Red Deer (*Cervus Elaphus Scoticus*) is a subspecies of Red Deer and lives in British Isles. The males roar loudly and repeatedly during the breeding season and females prefer a high to a low roaring rate. The males want to increase their territory and the number of hinds in their harems. So, the course of fight is unavoidable. Although it is possible that a male has no territory and harem, hence, they prefer to mate with a handy hind. In a nutshell, RDA starts with an initial population, called Red Deers (RD). They are divided into two types: hinds and male RDs. Besides, a harem is a group of female RDs, and the competition of male RDs to get the harem with more hinds via roaring and fighting, and their mating behavior is the basis of the proposed evolutionary algorithm. In continuous, the steps of the algorithm are detailed.

3.1. Initialization

Like other metaeuristics, an array is considered to be optimized. For example in GA, the terminology of array is named "*chromosome",* but here, in RDA, this term, "*Red Deer*" is used for this array. Therefore, Red Deer is the counterpart of a solution. In an N_{var} –dimensional optimization problem, a Red Deer is a $1\times N_{var}$ array. This array is defined by:

$$Red\ Deer=[X_1, X_2, X_3, \dots ,X_{N_{var}}]. \tag{9}$$

Also the value of function can be evaluated for each RDs as follows:

$$\text{Value}=f(Red\ Deer)=f(X_1, X_2, X_3, \dots ,X_{N_{var}}). \tag{10}$$

To start the algorithm, the initial population of size N_{pop} is generated. Some of the best RDs are selected to N_{male} and the rest of to N_{hind}.

3.2. Roaring male RDs

In this step, the male RDs are trying to increase their grace by roaring. Therefore, as happened in nature, the roaring process may be successful or faces with failure. In solution space, the neighbors of the good solutions are found (male RDs) and if the objective functions of the neighbors are better than the good solutions, they replace with the prior ones. In fact, every male RD is permitted to change their position.

3.3. Selecting γ percent of best males as commanders

In nature, there are so differences between male RDs. Some of them are more powerful, more attractive, or more successful in territory expanding than the others. So, RDs are divided into two types: commanders and stags, and the number of commander males is computed as follows:

$$N_{Com}=round\{\gamma . N_{male}\} \tag{11}$$

where N_{Com} is the number of males that form the harems. So the number of stags, is calculate by:

$$N_{stag}=N_{male}-N_{Com} \tag{12}$$

where N_{stag} is number of stags in males population.

3.4. Fighting between commanders and stags

User lets each commander fight with stags randomly. In solution space, a commander and a stag are approached to each other. So, the two new solutions are obtained and replaced the commander by the better solution, i.e. has better objective function among four solutions (commander, stag, and two new solutions obtained after approaching).

3.5. Forming harems

Now, the harems are formed. A harem is a group of hinds that a male commander seized them. The number of hinds in harems, depend on the power of male commanders. To form the harems, hinds among commanders are divided proportionally as follows:

$$V_n = v_n - \max_i \{v_i\} \tag{13}$$

where v_n is the power of nth commander and V_n is its normalized value. Having the normalized powers of all commanders, the normalized power of each male commander is calculated by:

$$P_n = \left| \frac{V_n}{\sum_{i=1}^{N_{Com}} V_i} \right| \tag{14}$$

From another point of view, the normalized power of a male commander is the portion of hinds that should be possessed by that male. Then the number of hinds of a harem will be:

$$N.harem_n = round\{P_n \times N_{hind}\} \tag{15}$$

where $N.harem_n$ is the number of hinds in n^{th} harem and N_{hind} is the number of all hinds. To divide the hinds to each male commander, user randomly chooses $N.harem_n$ of the hinds. These hinds along with the male will form n^{th} harem.

3.6. Mating commander with α percent of hinds in his harem

Like other species in nature, deers mate with each other. This action is done by a commander and $\propto$ percent of hinds in his harem are the parents.

$$N.harem_n^{mate} = round\{\propto \times N.harem_n\} \tag{16}$$

where $N.harem_n^{mate}$ is the number of hinds of n^{th} harem that mate with their commander. In solution space, user randomly chooses $N.harem_n^{mate}$ of the $N.harem_n$.

3.7. Mating commander with β percent of hinds in another harem

A harem is selected randomly (name it k) and let male commander mate with β percent of hinds in this harem. In fact, the commander attacks to another harem to expand his territory. The number of hinds in the harem which mate with the commander is computed as follows:

$$N.harem_k^{mate} = round\{\beta \times N.harem_k\} \tag{17}$$

where $N.harem_k^{mate}$ is the number of hinds in k^{th} harem which mate with the commander.

3.8. Mating stag with the nearest hind

In this step, each stag mate with its closest hind. In breeding season, the male RDs prefer to follow the handy hind. This hind may be his favorite hind among all hinds without consideration of harem territories, i.e. this hind maybe in his harem or habituates in another harem. User lets each stag mate with the nearest hind. To find the nearest hind, we need to figure the distances for each stag and all hinds. The distance between a stag and all hinds in *J*-dimension space, is calculate as follows:

$$d_i = \left(\sum_{j \in J} (stag_j - hind_j^i)^2\right)^{1/2} \tag{18}$$

3.9. Selection the next generation

To form the next generation, the two different strategies are followed. At first, user keeps all the male RDs, all commander and all stags, i.e. a percent of best solutions out of all solutions. For the remainder of population in the next generation, user chooses hinds out of all hinds in this generation according to their fitness by tournament, roulette wheel, or other evolutionary selection mechanism.

3.10. Stopping condition

The stopping condition may be the number of iteration, the quality of the best solution ever found, or a time interval. The pseudo-code of the RDA is shown in **Fig. 1**.

```
Initialize the Red Deers population.
Calculate the fitness and sort them and form the hinds ($N_{hind}$) and male RDs ($N_{male}$).
X*=the best solution.
While (t< maximum number of iteration)
    for each male RDs.
        A local search near his position.
        Update the position if better than the prior ones.
    end for
    Sort the males and also form the stags and the commanders (Eq. 11 & Eq. 12).
    for each male commanders
        Fight between male commanders and stags.
        Update the position of male commanders and stags
    end for
    Form harems (Eq. 13, Eq. 14 & Eq. 15)
    for each male commanders
```

```
        (Eq. 16)
        Mate male commander with the selected hinds of his harem randomly.
        Select a harem randomly and name it k.
        (Eq. 17)
        Mate male commander with some of the selected hinds of the harem.
    end for
    for each stags
        Calculate the distance between the stag and all hinds and select the nearest hind (Eq. 18).
        Mate stag with the selected hind.
    end for
    Select the next generation with roulette wheel selection.
    Update the X* if there is better solution.
    t=t+1.
end while
return X*
```

Fig. 1. The pseudo-code of the RDA

4. Experimental results

In this section, some experiments are design to evaluate the algorithms. In this way, RDA is firstly used to handle the proposed problem. In addition, to compare the proposed method, Genetic algorithm as a well-known evolutionary algorithm [11] and Imperialist Competitive Algorithm as a successful recent method are utilized [12].

Here, an experiment to analyze the performance of the algorithms is designed. The number of cities and vehicles in each echelon directly indicate the complexity of problem. Hence, the problems are divided into three levels (i.e. small, medium and large). In each level, three random problems are generated.

The parameters of algorithms are tuned with full factorial and Taguchi experiments [13]. And final value for each parameter for all algorithms is shown in **Table 1**.

Table 1. Parameters of the algorithms

Algorithms	Parameters
GA	Initial population=100, Maximum number of iteration=200, Mutation rate=0.2, Mutation percentage=0.2, Crossover rate=0.8, Crossover percentage=0.8
ICA	Initial population=100, Maximum number of iteration=200, Number of empires=15, Number of colonies=Initial population-Number of empires, Assimilation coefficient=2, Revolution probability=0.2, Rate of revolution=0.1, Colonies mean cost coefficient=0.1
RDA	Initial population=100, Maximum number of iteration=200, Number of males=15, Number of hinds=Initial population-Number of males, Alpha=0.9, Betta=0.4, Gamma=0.7

To show the performance our proposed method, the results are given in **Table 2.** As shown in this table, however the RDA has reached the better value and found the better solution in each classification of problems. But, it needs more time. Besides, an analysis for affection of the problem size and three proposed algorithms is depicted in **Fig. 2**. This figure shows the interaction between the quality of the presented algorithms and the size of problems. It is obviously that RDA exhibit robust performance, while the problem size increases.

Table 2
The results of Vehicle Routing Problem (VRP)
Sol= Solution, CT= CPU Time (Sec)

Type of problems	No. problem	No. Cities	No. Vehicles	GA		ICA		RDA	
				Sol	CT	Sol	CT	Sol	CT
small	1	8	3	354.4722	14.887684	354.4722	16.660796	354.4722	15.36149
	2	10	3	468.4243	13.091802	506.4388	18.273574	461.9244	17.96314
	3	14	4	808.2543	20.986772	534.0541	25.500430	522.785	23.166792
medium	4	20	4	667.3689	21.229507	651.4673	25.692272	594.0197	29.809790
	5	25	5	1187.3328	37.868352	815.8704	47.935897	794.7334	48.188497
	6	30	5	1026.0682	32.838265	998.7305	45.187324	975.1149	44.045288
large	7	40	6	1403.9785	39.804141	1642.2008	54.044008	1367.1802	68.023085
	8	50	7	1608.58	63.344477	1861.3722	82.504280	1584.5904	73.390045
	9	60	7	1859.7725	69.124553	2319.0633	87.488507	1748.1033	92.259476

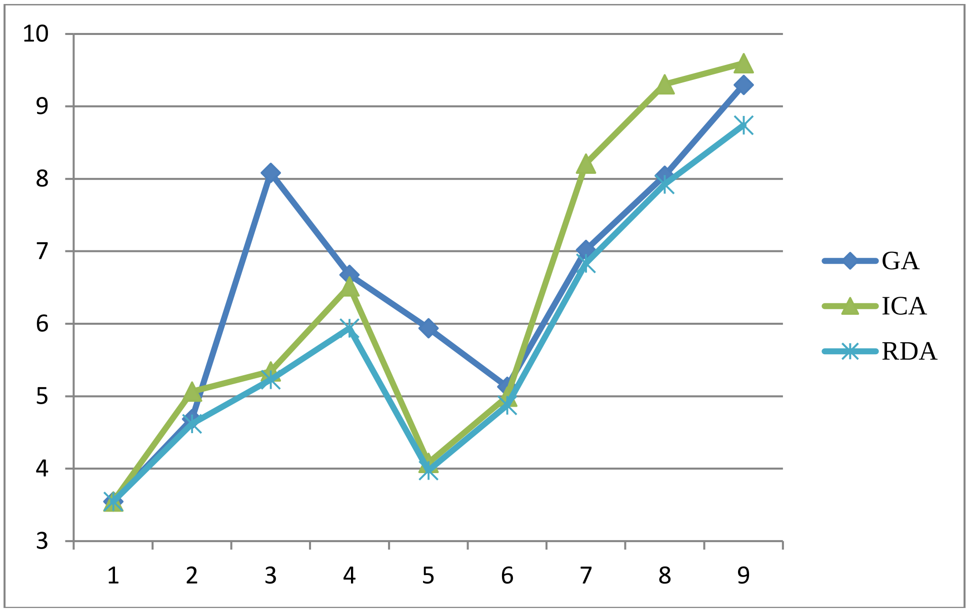

Fig. 2. Means plot for the interaction between each algorithm and problem size for VRP

5. Conclusion and future works

In this work, the vehicle routing problem is used to evaluate the Red Deer Algorithm (RDA) for the first time. In this way, two powerful methods: GA and ICA are utilized to compare the results. All of parameters are tuned by full factorial and Taguchi experiments. The problems are classified in three levels to explore the algorithms exactly. The results show the performance and efficiency of RDA in this type of transportation problem.

For more studies, researchers can modified some parts of algorithm to improve the execution speed. In addition, one of important advantages of proposed algorithm is balance between the intensification and the diversification phases by intelligent ways. Hence, more comprehensive analysis by other real scale optimization problems may steel needed to be investigated.

References:

[1] Barrie M. Baker, M. A. Ayechew, “A genetic algorithm for the vehicle routing problem.” *Computers and Operations Research*, 30, (2003), 787–800.

[2] T. H. Clutton-Brock, S. D. Albon, R. M. Gibson, “the logical stag: Adaptive Aspects of fighting in Red Deer.” *Animal Behavior*, 27, (1979), 211-225.

[3] K. E. McComb, “Female choice for high roaring rates in Red Deer, Cervus elaphus.” *Animal Behavior*, 41, (1991), 79-88.

[4] B.L. Golden, "Vehicle routing problems: formulations and heuristic solution techniques." *ORC technical report 113*, Massachusetts Institute of Technology (August, (1975)).

[5] B. Bullnheimer, R. F. Hartl, C. Strauss, “An improved ant system algorithm for the vehicle routing problem.” *Annals of Operations Research*, 89, (3), (1999), 19–28.

[6] D. T. Hiquebran, A. S. Alfa, Shapiro, D. H. Gittoes, “A revised simulated annealing and cluster-2rst route-second algorithm applied to the vehicle routing problem.” *Engineering Optimization*, 22, (77), (1994), 107.

[7] Y. Zhou, Q. Lou, J. Xie, H. Zheng, “A hybrid Bat Algorithm with Path Reliking for the Capacitated Vehicle Routing Problem.” *Metaheuristics and Optimization in Civil Engineering.* Springer International Publishing, (2016), 255-276.

[8] B. Sarasola, K. F. Doerner, V. Schmid, E. Alba, “Variable neighborhood search for the stochastic and dynamic vehicle routing problem.” *Annals of Operations Research*, 236, (2), (2016), 425-461.

[9] I. H. Osman, “Meta strategy simulated annealing and Tabu search algorithms for the vehicle routing problem.” *Operations Research*, 41, (4), (1993), 21–51.

[10] A. M. Fathollahi Fard, M. Hajiaghaei-Keshteli, “Red Deer Algorithm (RDA); A New Optimization Algorithm Inspired by Red Deers’ Mating.” *12th International Conference on Industrial Engineering (ICIE 2016)*, Tehran, Iran, (2016), 34-35.

[11] M. Hajiaghaei-Keshteli, “The allocation of customers to potential distribution centers in supply chain network; GA and AIA approaches.” *Appl. Soft Comput*.11, (2011), 2069–2078.

[12] E. Atashpaz-Gargari, C. Lucas, “Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition.” in: *IEEE Congress on Evolutionary Computation*, Singapore (2007), 4661–4667.

[13] G. Taguchi, “Introduction to quality engineering.” *White Plains, Asian Productivity Organization,* UNIPUB, (2000).