International Journal of Computer Applications Technology and Research
Volume 5– Issue 8, 519 - 524, 2016, ISSN:- 2319–8656

A Multi-Objective Optimization Algorithm to Predict Information in Mobile Databases

Leila Alaei Sheini
Department of Computer
Engineer, Khomein Branch,
Islamic Azad University
Khomein, Iran

*Hamid Paygozarh
Department of Computer
Engineer, Khomein Branch,
Islamic Azad University
Khomein, Iran

Mohammad Khalily Dermany
Department of Computer
Engineer, Khomein Branch,
Islamic Azad University
Khomein, Iran

Abstract: Recent advances in wireless data networking, satellite services and cell connections allow us to take advantage of mobile computing systems. Here, a problem is the mobility of computing process which needs to get information from databases. In fact, system movement during the transaction interrupts the mobile network connection to server wireless systems, and the disconnection causes failing in the running transaction so the mobile system has to run the transaction again. One way to avoid transaction failure is doing prediction process until connecting to the next wireless network. The objective of this study is to use particle swarm optimization (PSO) algorithm in order to provide a solution for prediction of the information sent to the mobile system for using the memory. Generally, PSO algorithms result in the best or an acceptable solution for optimized selection and design in many scientific and technological problems. Here, the objective of PSO is to select the best information to send to the mobile system when the connection is interrupted in order to prevent transaction failures. This study focuses only on information sent to the mobile system.

Keywords: mobile database, cache operations, mobile customer, cell connection, particle swarm optimization, multi-objective.

1. INTRODUCTION

Recent technical advances in portable computers as well as the wireless technology led to the emergence of portable computers with wireless data networking. Such systems allow users to act in distributed computing environments even if moving. Mobile users needs access to private or cooperative databases stored in mobile or fixed hosts and provide queries and updates of information for wired and wireless networks [1].

Such services include constraints such as limited power supply in mobile system, expensive services, limited bandwidth in mobile systems, and limited service area. Limited service area causes interruption in mobile network connection to server wireless systems which leads to request rejection in the running transaction. This makes the system rerun the process which is costly and time-consuming [2].

To avoid request rejection, an approach is to use cache operation in the mobile system. When receiving a request from the mobile system, mobile service provider predicts the information surplus to what is required by the request and sends it to the mobile system. The mobile system stores the surplus information into cache memory. If the data requested by the current or the next transaction is available in the cache memory, the mobile system will use it. In this way, there is no need to a connection to send the request and receive information. This prevents the failure of the running transaction while reducing costs and response time. The most important issue in this approach is accurate and optimized prediction of data which requires forecasting information available on server database. To this end, this paper takes advantages of the multi-objective particle swarm optimization algorithm [4].

The rest of this paper is structured as follows: Section 2 defines the mobile database; Section 3 studies PSO algorithm; the proposed algorithm is provided in Section 4; and the proposed algorithm is evaluated in Section 5.

2. MOBILE DATABASE

Traditionally, we had large-scale commercial databases that were developed as centralized database systems. However, this trend changed as more and more distributed applications started to emerge. Distributed database applications usually involved a strong central databases and powerful network administrations. However, trends of new technologies experienced changes due to emerging technologies such as:

- The notebook and laptop computers are being used increasingly among the Business Community

- The development and availability of a relatively low-cost availability of wireless digital communication infrastructure.

The rapid development of wireless communication and computer miniaturization technologies enables users to use computer resources anywhere on the computer network. For example, you can connect the World Wide Web even in-flight. Mobile database is the database that allows the development and deployment of database applications for handheld devices, thus, enabling relational database based applications in the hands of mobile workers. Mobile database technology allows employees using handled devices to link to their corporate networks, download data, work offline and then connect back to the network to synchronize with the corporate database. For example, with a mobile database embedded in a handheld device, a package delivery worker can collect signatures after each delivery and send the information to the database at the end of the day.

2.1 Defining the Mobile Database

A mobile database is a database that can be connected to by a mobile computing device over a mobile network so that the client/server connection is of wireless type. It also contains a cache to store data and transactions in disconnections. Database is a structure used to organize information, this information may be a list of contacts, prices, traveled distances and so on.

Laptops, mobile phones, and pocket PCs are widely used and probably we will face with the increasing number of mobile systems applications in the future. Although analysts cannot accurately predict the most popular and most applicable application of the future, it is quite clear that a significant percentage of applications will require the use of the database. Many future applications will require the ability of downloading information from the database and perform operations on data even if located out of the target range or the connection is interrupted.

Today, with the advent of mobile databases, users can remotely exchange information with the mobile databases using smart phones or PDAs without worrying about time or distance. Mobile databases allow employees to enter information into databases even in flight. Information can be synchronized with the database server after a short time.

2.2 Mobile Database Requirements

Some factors that shall to be considered in mobile processing include:

- Mobile users should be able to continue their activities without a wireless connection (either because of poor communication or even when disconnected).
- User-time is the most valuable factor in most business applications.
- Since the connection is costly, reducing the connection time as much as possible should be highly considered.
- The number of transmitted bytes or packets is another factor that shall be considered in calculations.
- Applications interactions should be significant.
- Applications should be able to access local devices/hardware such as printers, bar code scanners, and GPS units (for mapping or navigator systems).
- The Bandwidth should be reserved (the minimum bandwidth required by wireless networks is a few megabytes)
- The user does not need access to data history, only the last modified data is required.
- A right time must be selected to update the information (during peak or off-peak periods)
- The limited lifetime of the power supply (battery)
- Changing the network topology

2.3 Mobile Database Systems Architecture

In any architecture, the followings shall be considered:

- Users are not located on a fixed geographical location;
- Mobile computing devices must be low power, low cost, and portable;
- Wireless networks;
- Computer limits;
- The mobile database structure

A mobile database normally consists of three components:

- Fixed host
- Mobile units
- Base stations

A mobile device is a device with the ability to connect to a fixed network through wireless connection. In fact, mobile unit is a communication device that operates with batteries and can move in a certain limited geographical area. The geographical area is limited due to limited bandwidth in wireless communication channels.

Fixed hosts perform the transaction and data management functions with the help of data base servers (DBS). Mobile units are portable computers that move around the geographic area that includes mobile network and are used for connecting to the base stations (note that mobile networks are not necessarily cell phone networks).

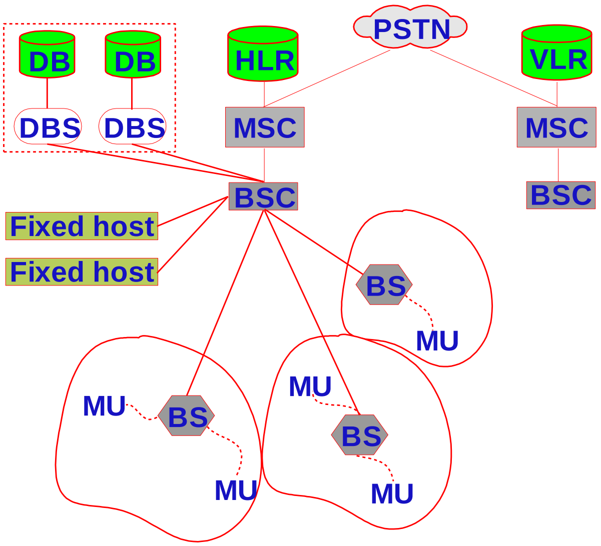


Figure 1. Transaction management in mobile database

Base stations are two-way radios installed at a fixed location establishing communication channels from the mobile unit to the fixed host and vice versa. They are usually low-power devices such as mobile phones and wireless routers.

3. PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO) is a global optimization algorithm for dealing with problems in which a best solution can be represented as a point or surface in an n-dimensional space. Hypotheses are plotted in this space and seeded with an initial velocity, as well as a communication channel between the particles. Particles then move through the solution space, and are evaluated according to some fitness criterion after each time-step. Over time, particles are accelerated towards those particles within their communication grouping which have better fitness values. The main advantage of such an approach over other global minimization strategies is that the large number of members that make up the particle swarm make the technique impressively resilient to the problem of local minima. Fig. 2 shows examples of particle movement in the search space. In Fig 2, the top left image represents the initial position of particles in a two-dimensional search space, and after some iterations, particles eventually converge as the down-right image.

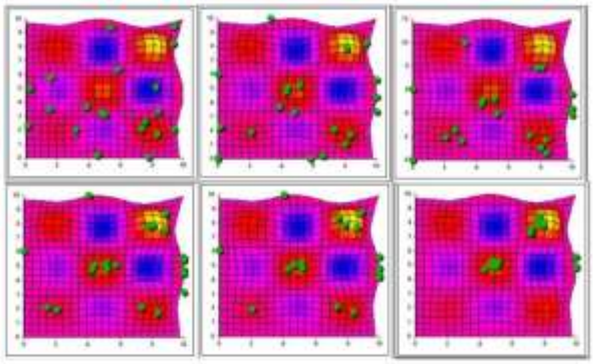


Figure 2 Movement of particles in a group [8]

Each particle has a position representing the coordinates in the multi-dimensional search space. The position of the particle changes as it moves. Here, xi (t) denotes the position of particle i at time-step t and vi (t) is the velocity of the particle i at time-step t. By considering the velocity, the current position of each particle can be calculated as Eq. 1.

$$\mathrm{x_i}(\mathrm{t}+1)=\mathrm{x_i}(\mathrm{t})+\mathrm{v_i}(\mathrm{t}+1) \\ x_i(t) \sim U(x_{min}, x_{max}) \qquad (1)$$

The fitness of the position of a particle in the search space is assessed through a fitness function. Particles have the ability to remember the best position during their lifetime. The best position personally found by a particle is yi (in some algorithm yi is also called pbest). In optimization process, the velocity vector reflects the particle empirical knowledge and the society information. Each particle moving in the search space considers two components:

- Cognitive component: $\mathrm{y_i}(\mathrm{t})-\mathrm{x_i}(\mathrm{t})$ is the best solution personally found by a particle.
- Social component: $\hat{\mathrm{y}}_\mathrm{i}(\mathrm{t})-\mathrm{x_i}(\mathrm{t})$ is the best solution found by the group.

Two basic models are introduced for standard PSO algorithm: calculating the velocity vector based on cognitive and social components.

4. THE PROPOSED ALGORITHM

To solve an optimization problem, we must first define the position of each particle. In this problem, the position of each particle is considered as a [records number-table number] set. In other words, each particle is a set of records. The particle length is variable and is calculated as the difference between the user's request and the network packet length. Table 1 shows an example of a position for a request set.

Table 1 Set of particles for the proposed record set

Rec i-Table j		Rec i-Table j		....	Rec i-Table j		Fitness
1	65	4	245	....	3	567	

Each particle has a table number and a record number.

4.1 Generating the Initial Population

Here, 50% of the initial population is randomly selected and the remaining 50% using an intelligent approach based on the maximum entropy. Entropy is the maximum difference since it generates a fitted population. Entropy is calculated based on Eq. 2:

$$\max \sum_{\forall\ 1 \le i,j \le n \ i \ne j} (|a_i - a_j|) \qquad (2)$$

At this stage, population size (the number of particles should exist in the population pool) is determined. This means that population size is an important factor affecting the algorithm efficiency. If its size is too small, only a small part of the solution space will be searched and the solution will quickly converge to probably a local optimum. And if the population size is very large, time-consuming calculations will be needed that is disproportionate to the solution. In Section 5 (algorithm evaluation), different population sizes are examined and the right size is found.

4.2 Calculating the Fitness Function

The fitness function evaluates the fitness and the performance of each particle of the population in problem-solving. The structure of each row is shown in Table 2.

Table 2: The structure of a row

Row A	
Table Code	TC
Record Code	RC
Last Record Sent Time	TR
Request Weight	WR
Usage Weight	WU
The Final Vote	Vote

So that:

Table Code (TC): Here, an ID is defined for each table in the database structure, and TC represents the table whose record is used.

Record Code (RC): According to the definitions of databases, the unique ID called key shall be assigned to each record (or sample entities). Here, RC is used.

Last Record Sent Time (TR): It is the last time that RC is sent from TC table to the mobile unit by the agent record. TR shows the agent which records have been (or have not been) recently used in the database. Thus, the agent removes the records that have not been long used since they probably are not interested by A.

Usage Weight (WU): It counts the times that RC from TC table is used by the mobile unit and modified or updated. The difference between WR and WU is that a user may request a record but does not change it.

Vote: Using the profile information, the agent can specify the most-used records (according to the request time and the weight). To this end, a coefficient (an importance factor which is calculated by trial and error) is considered for the mentioned variables as:

- Sent time × 0.65
- Request weight × 0.25
- Usage weight × 0.10

The Vote for ith record is calculated as follows:

$$\mathrm{Vote}_{(\mathrm{i})} = (\mathrm{Tr} \times 0.65) + (\mathrm{Wr} \times 0.25) + (\mathrm{Wu} \times 0.10) \qquad (3)$$

To solve the problem of mobile information prediction, we need to define a function measuring the fitness value for the obtained solutions. To measure the fitness of each particle, we can calculate Vote for each record, then consider the mean Vote as the fitness function.

4.3 Defining the Algorithm Parameters

4.3.1 Particles Initialization

The first step in a PSO algorithm is to initialize the swarm and control parameters. Usually, the particle's position is uniformly distributed to cover the search space. It should be noted that PSO performance is affected by the initial distribution of particles in the swarm (i.e. how much of the search space is covered and how the particles are distributed in the search space). If the optimal solutions are located somewhere in the search space that is not covered by the initial swarm, PSO will face difficulties to find them. In this case, PSO can find the optimal solutions only if the particles movements guide the search process toward these uncovered solutions.

Position initialization for each particle is defined as Eq. 4.

$$x(0) = x\_(min,j) + r\_j\ (x\_(max,j) - x\_(min,j)\ ), \forall j = 1,2 \qquad (4)$$

Where xmin and xmax denote the minimum and maximum values for both table number and row number, respectively. Here, rj ~ U (0, 1).

4.3.2 Velocity C

To decide the efficiency and accuracy of a PSO algorithm, an important factor is the ability to achieve an optimal compromise between Explore and Exploit. Exploration is the ability to search for new solutions far from the current solution in the search space. On the other hand, exploitation tis he ability to concentrate the search around a promising area to refine a candidate solution. We found a solution achieving an optimal compromise between these two conflicting goals through PSO velocity updating which is shown in Eq. 5.

$$v_{ij}(t+1) = v_{ij}(t) + c_1 r_{1j}(t)\left[y_{ij}(t) - x_{ij}(t)\right] + c_2 r_{2j}(t)[\tilde{y}_j(t) - x_{ij}(t)]$$
$$v_{ij}(t+1) = \begin{cases} v_{ij}(t+1) & if\ v_{ij}(t+1) < v_{max,j} \\ v_{max,j} & if\ v_{ij}(t+1) \geq v_{max,j} \end{cases} \qquad (5)$$

where $v_{max,j}$ is the maximum allowed velocity in dimension j. The value of $v_{max,j}$ is very important since it guides the search by clamping the velocity. Large values of $v_{max,j}$ facilitate global exploration while small values encourage local exploitation. If $v_{max,j}$ is too small, the swarm may not explore sufficiently beyond locally good regions. Furthermore, the swarm may become trapped in a local optimum with no means of escape. On the other hand, too large values of $v_{max,j}$ are associated with the risk of missing a good region. The particles may jump over good solutions, and continue to search in fruitless regions of the search space. While large values do have the disadvantage that particles may jump over optima. In this case the particles move faster.

This leaves the problem of finding a good value for $v_{max,j}$ in order to balance between:

- Moving too fast or too slow
- Exploration and exploitation

Usually, the $v_{max,j}$ values are selected to be a fraction of the domain of each dimension of the search space. That is:

$$v_{max,j} = \delta(x_{max,j} - x_{min,j}) \qquad (6)$$

where $x_{max,j}$ and $x_{min,j}$ are respectively the maximum and minimum values in dimension j, and δ∈ (0,1]. The value of δ is equal is first equal to 1 and varies in every generation according to Eq. 7. In each generation, the value of δ is 90% less than the previous generation.

$$\delta = 0.9^i \quad \text{i = generation number} \qquad (7)$$

4.3.2.1 Algorithm Stopping Condition

The stopping condition is defined based on swarm radius. The algorithm terminates when the normalized swarm radius is close to zero. Normalized swarm radius is calculated as:

$$R_{norm} = \frac{R_{max}}{diameter(S)} \qquad (8)$$

where diameter (S) denotes the diameter of the initial swarm and R_{max} is maximum radius calculated as:

$$R_{max} = \|x_m - \hat{y}\ \| \quad , \qquad m = 1, \dots, n\_s \qquad (9)$$

The algorithm stops when R_{norm} is close to zero. Here, $\hat{y}$ is the minimum value of the fitness function.

4.3.3 Structure of the Proposed Algorithm

- Step 1: Initializing particles velocity and position in initial swarm according to Eq. 4;
- Step 2: Updating the best local position of ith particle for all particles;
- Step 3: Updating the best global position of all particles;
- Step 4: Calculating the new velocity of all particles using Eq. 5;
- Step 5: Calculating the new position of all the particles using Eq. 10.

$$x\_i\ (t+1) = x\_i\ (t) + v\_i\ (t+1) \qquad (10)$$

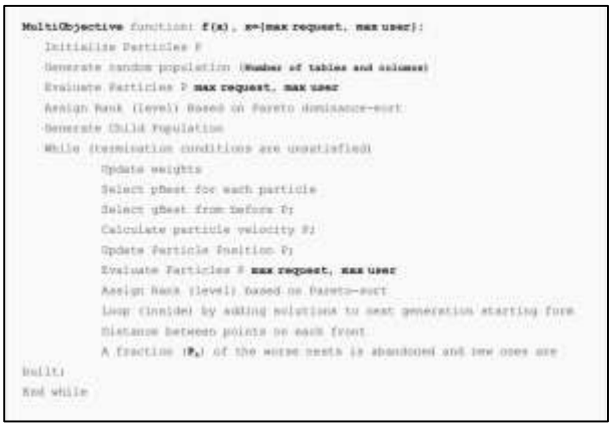
- Step 6: Repeating steps 2 through 5 considering the stopping condition discussed Section 4.3.3.

4.4 Multi-Objective Solutions

A drawback of the proposed method is failing to consider the previous needs of the user based on the vote, since it considers the overall needs of all users. To tackle this, the problem is defined as a multi-objective problem simultaneously considering needs of all users and needs of the current user. For this multi-objective algorithm, the code structure shown in Fig. 3 is proposed.

5. EVALUATION OF THE PROPOSED ALGORITHM

An important issue for a theory is testing and proving it. To test the presented theory, we developed codes in MATLAB environment which will be discussed here. The tests were conducted using a system with 4GH processor and 6GB memory. The data used in tests included information obtained from Sama University of Mahshahr.



```
MultiObjective function: f(x), x=[max request, max user]:
    Initialize Particles P
    Generate random population (Number of tables and columns)
    Evaluate Particles P max request, max user
    Assign Rank (level) Based on Pareto dominance-sort
    Generate Child Population
    While (termination conditions are unsatisfied)
            Update weights
            Select pBest for each particle
            Select gBest from before P;
            Calculate particle velocity P;
            Update Particle Position P;
            Evaluate Particles P max request, max user
            Assign Rank (level) based on Pareto-sort
            Loop (inside) by adding solutions to next generation starting form
            Distance between points on each front
            A fraction (Pa) of the worse nests is abandoned and new ones are
built;
End while
```

Figure 3: the pseudo code proposed for multi-objective PSO

We ran the algorithm with initial populations of 125, 250, 500, and 1000. The algorithm was evaluated based on the processing time, as shown in Fig. 4.

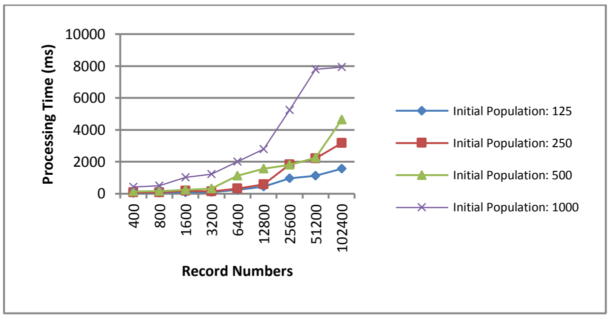


Figure 4 Processing time vs. record numbers for different initial populations

As seen in Fig. 4, the initial population of 125 is the optimal population in terms of running time.

Then, we run the algorithm for 200 users with 20 buffer. The results are listed in Table 4 and Fig. 5.

Table 4 Prediction percentage for 200 users with 20 buffer in the mobile unit and in terms of record number

	Number of Users = 200 Number of buffer = 20				
Total Record	200	400	800	1600	3200
Total Requests	2108	4419	5140	6703	8615
Predicted Records	775	1699	2084	2781	3943
Predicted Percent	36.76%	38.45%	39.84%	41.49%	45.77%

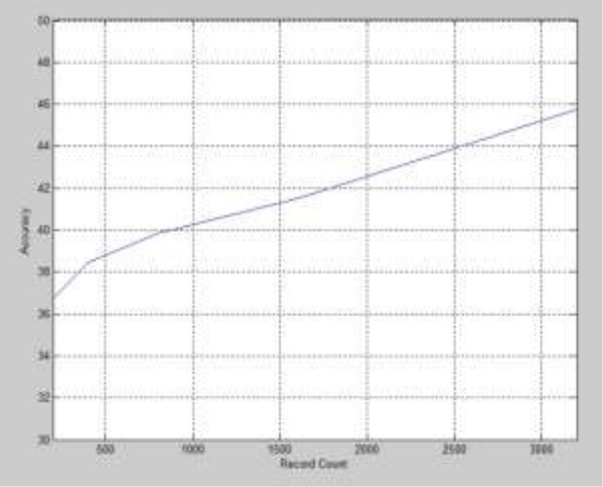


Figure 5 Prediction percentage for 200 users with 20 buffer in the mobile unit

5.1 Comparison with Other Algorithms

To compare the proposed algorithm with other algorithms, we applied the genetic algorithm using MATLAB. We also compared the algorithm with the results obtained in [3]. The output is shown in Table 5. Since we did not have access to standardized data, we evaluated the algorithms based on the prediction accuracy.

Table 5 Comparison between the proposed algorithm, genetic algorithm, and the proposed algorithm in [3]

	Algorithm	Accuracy
1	Proposed Algorithm	45.77%
2	Genetic Algorithm	35.76%
3	Proposed Algorithm in [3]	37.86%

6. CONCLUSIONS

In studies mentioned above, the only cashed information is the one that is now required, and there is no prediction for the information that may be used in the future. This study employed particle swarm optimization algorithm. To evaluate the proposed algorithm, some tests were conducted and the results indicated an efficiency of 25 to 45 percent for predictions that is reasonable.

7. REFERENCES

[1] Tarafdar M, Haghjoo M. Processing Spatial-Prediction Queries in Mobile Database using Timed Sequential Patterns. Tabriz; 2011. p. 234-41.

[2] Mohammadi M, Habibi A, Mohammadi F. Using Fuzzy Database to Predict Data and Improve Cache Operations in Mobile Databases. Science and Research University, East Azerbaijan; 2011.

[3] Beigi H. Investigation Transactions in Mobile Database and Using Genetic Algorithm to Predict Data in Cache Operations. MS's Thesis, 2008.

[4] Xuan K, Zhao G, Taniar D, Rahayu W, Safar M, Srinivasan B. Voronoi-based Range and Continuous Range Query Processing in Mobile Databases. J Comput Syst Sci. 2011; 77(4):637–51 .

[5] Srikant R, Agrawal R. Mining Sequential Patterns: Generalizations and Performance Improvements [Internet]. Springer; 1996 [cited 2014 Oct 2]. Available from: http://link.springer.com/chapter/10.1007/BFb0014140

[6] Fernando N, Loke SW, Rahayu W. Mobile Cloud Computing: A survey. Future Gener Comput Syst. 2013;29(1):84–106 .

[7] Yavaş G, Katsaros D, Ulusoy Ö, Manolopoulos Y. A Data Mining Approach for Location Prediction in Mobile Environments. Data Knowledge. 2005; 5 4(2):121–46 .

[8] Goldberg DE. Genetic Algorithms in Search, Optimization, and Machine Learning. 1 edition. Reading, Mass: Addison-Wesley Professional; 1989. 432 p .

[9] Mitchell M. An Introduction to Genetic Algorithms. Reprint edition. Cambridge, Mass.: A Bradford Book; 1998. 221 p .

[10] Memetic algorithm [Internet]. Wikipedia, the free encyclopedia. 2014 [cited 2014 Oct 28]. Available from: http://en.wikipedia.org/w/index.php?title=Memetic_algorithm&oldid=626348976

[11] Moscato P, Cotta C, Mendes A. Memetic algorithms. New Optimization Techniques in Engineering [Internet]. Springer; 2004 [cited 2014 Oct 28]. p. 53–85. Available from:http://link.springer.com/chapter/10.1007/978-3-540-39930-8_3

[12] Neri F, Cotta C, Moscato P, editors. Handbook of Memetic Algorithms. 2012 edition. S.l.: Springer; 2013. 370 p .

[13] Euclidean [Internet]. Wikipedia, the free encyclopedia. 2014 [cited 2014 Oct 28]. Available from: http://en.wikipedia.org/w/index.php?title=Euclidean&oldid=630468632