New Hybrid Algorithm for Task Scheduling in Grid Computing to Decrease missed Task

Z. Pooranian, A. Harounabadi, M. Shojafar and N. Hedayat

Abstract**—**The purpose of Grid computing is to utilize computational power of idle resources which are distributed in different areas. Given the grid dynamism and its decentralize resources, there is a need for an efficient scheduler for scheduling applications. Since task scheduling includes in the NP-hard problems various researches have focused on invented algorithms especially the genetic ones. But since genetic is an inherent algorithm which searches the problem space globally and does not have the efficiency required for local searching, therefore, its combination with local searching algorithms can compensate for this shortcomings. The aim of this paper is to combine the genetic algorithm and GELS (GA-GELS) as a method to solve scheduling problem by which simultaneously pay attention to two factors of time and number of missed tasks. Results show that the proposed algorithm can decrease makespan while minimizing the number of missed tasks compared with the traditional methods.

Keywords**—**Grid Computing, Genetic Algorithm, Gravitational Emulation Local Search (GELS), missed task

I. INTRODUCTION

GRID computing systems have become popular for the resolution of larg-scale complex problems from science, engineering, finance, etc. As larg-scale infrastructures for parallel and distributed computing systems, grids enable the virtualization of a whole range of resource, despite their high degree of heterogeneity. Thus, different type of grid systems have been define. Such systems are currently comprising computational grids, desktop grids, enterprise grids, scavenging grids, data grids, etc. In fact, computational grids(CGS) were the first to address the resolution of complex problems from e-science. They are also useful in solving the problems, arising in meterology, industry, physics, medicine, finance, etc, for witch grid-enabled solutions have made possible to achieve breakthroughs in their resolution time[1].

In order to accomplish a job requiring larg-scale computation over a distributed systems, it is difficult to have afair judgment on which dispatching method leads to optimal solution. However, mapping independent tasks on to a heterogeneous computing(HC) suite is a well-known NP-Complete problem[2].By regarding to this problem and grid extension and its dynamics, hence, deterministic algorithms can't involve appropriate efficiency for solving this problem. Therefore, much research was done on heuristic methods. The most of these methods try to minimize Makespan. Many heuristic algorithms were suggested for task scheduling in grid, such as: HSPN [3], Genetic Algorithm (GA) [4], Simulated Annealing (SA) [5], or Tabu Search (TS) [6]. Among them, Genetic Algorithm is the best heuristic method, because it is parallel inherently and it can search problem space in several aspects simultaneously. Since, convergence of Genetic Algorithm is slow to global optimization and also its instability has been proved in different implementation of algorithm which Genetic Algorithm efficiency can be improved by combination with the other algorithms. In this research, the combination of Genetic Algorithm and GELS. are used. Since, Genetic Algorithm is weak in local search and it is strong in global search and versus, GELS is a local search algorithm by imitation of gravitational attraction, so it is strong in local search and it is weak in global search. For solving grid scheduling problem the combination of benefits of these two algorithms was used. In this paper, static scheduling algorithm was presented for solving scheduling problem of independent tasks in grid. The word "Static Scheduling" means that all necessary data about tasks, resources, the number of resources, Should be specified before implementation. The advantage of being static is that no overhead is exerted on the system. In this algorithm, in addition to decreasing Makespan, the Quality of Service (QOS) was tried to be regarded that their missed tasks are decreased by finished deadlines. In the second section, previous related works in this field are described briefly. In the third section tasks scheduling problem was described. In the 4 and 5 sections, intelligent Genetic Algorithm and GELS algorithm, were described respectively. In fifth section the proposed algorithm is described in details. The proposed algorithm is compared with several similar algorithms, and then conclusion and the future works are mentioned in last section. In continuous, was described every section generally.

Z. Pooranian, Department of Computer Engineering, Dezful Branch, Islamic Azad University, Dezful, Iran (e-mail: Zahra.pooranian@ gmail.com).
A. Haroonabadi, Associate professor in Computer Eng- Software, Islamic Azad University, Central Tehran Branch, Department of computer, Tehran, Iran (e-mail: A.Harounabadi@gmail.com).
M. Shojafar, Msc. in Computer Eng- Software, Qazvin Islamic Azad University, Qazvin, Iran (e-mail: Shojafar@qiau.ac.ir).
N. Hedayat , Civil Engineering Department, Dezful Branch, Islamic Azad University, Dezful, Iran (e-mail: n.hedayat@yahoo.com).

II. RELATED WORKS

Genetic Algorithm in [7] was presented to solve the scheduling problem of dependent tasks in which the population quantity and the number of generation are depended on the number of tasks. Comparing to traditional Genetic Algorithm, this work which the number of its iteration is stable for any task, has the advantage if the number of tasks is low, so long computational time was not used and if the number of tasks is much the probability of finding the appropriate solutions is provided through further repetition of algorithm. Also, SA was used to decrease the time of calculations. In [8], Genetic Algorithm was presented for scheduling of independent tasks in which the heuristic initialization of initial population using MCT (Minimum Completion time) Algorithm that lead to shortening the search time and making the convergence faster. By regarding to the large capability of SA in finding local optimized conclusions. In [9] combination of local search algorithm were used for scheduling by using of SA and global search by genetic (GSA). In this approach, if in each generation, changed chromosomes by genetic operator don't improve comparing to previous generation they are affected by SA and are likely accepted for the next generation and this work lead to increase search efficiency, further convergence rate and ran of local minimum. In [10], some modification have been in Genetic Algorithm to improve the scheduling efficiency. These changes are consisted of the combination of Greedy algorithms with Genetic Algorithm to minimize tasks start-time until at the end, Makespan is minimized. Also it can use idle times of processors. In this algorithm, two fitness functions were used. In the first function, it was searched for chromosomes with the shortest Makespan and in the second section, it was researched for the most appropriate chromosomes in the respect of load balance. RFOH algorithm was presented in [11] which in addition to try for decrease Makespan, it is tried to increase fault tolerance comparing to previous methods. Thus, the failures in resources are stored in GIS until when broker has tasks for scheduling, it uses these data in fitness function of Genetic Algorithm until it can find resources in which the probability of faults incidence is less.In [12], the Genetic Algorithm is presented in which both QOS parameters including time and cost were regarded simultaneously and because these two parameters are in conflict each other and they can't improve together simultaneously and one improvement leads to efficiency decrease in the other, it gives weight to each parameter as the weighing is done by user as each of the parameters has more value for the user gives more weight and the other gives less weight, or weighting is happened randomly. In [13], the Genetic Algorithm is presented in which chaotic variables were used instead of random variables for chromosomes production. This work lead to distributing of solutions in the whole search space and avoid from local minimums, the best solutions and production are obtained in the shorter time. In [14], the Genetic Algorithm was combined with hill-climbing algorithm to repair the chromosomes. This work lead to change invalid individual in each generation until they become valid individual in new population. GELS algorithm in [15] was used for resource reservation and scheduling of independent task, hence, in objective function, if one resource can't perform a task in its defined deadline, it changes the resources and allocate task to the other resource for executing. Simulation conclusions show that this algorithm decreases makespan comparing to genetic. In previous methods, decreasing of the whole time of implementation was further regarded and the number of missed tasks and load balance problem beside time decreasing were not regarded in any ones. In proposed algorithm, it was tried to regard to those three parameters simultaneously. Also, regarding that genetic is weak in local searching and with combining it with local searching algorithms it was tried to improve its weak. In proposed algorithm, the combination of Genetic Algorithm and GELS algorithm was used by regarding that GELS algorithm is searches the space problem well and find better solutions comparing to the other local searching algorithms such as hill-climbing, SA &.

III. Scheduling Problem Description

The scheduling problem of independent tasks is a NP-hard problem that consist of N tasks and M machines. Each task should be processed by each M machine, as the Makespan is minimized. Here, we have just regarded to one of the QOS parameter which is time constraint and we have ignored the cost. In other word, we have introduced a Deadline (D) for every task as each task should end its implementation before ending D. Each task can be only implemented on one resource and it is not stopped before finalizing its implementation. We use the expected time to compute (ETC) matrix model which was described in [16]. Since the proposed scheduling algorithm is as static, we have supposed that expected implementation time for each task, i, on each resources j, was determined before and was set on ETC matrix, ETC[i, j]. Also, ready time (Ready[M]), is a time when the machine M has finished its previous task. Here, Makespan is equal to maximum complete time, Completion_Time[i,j](Equation 1).

$$\text{Makespan}=\text{Max}(\text{Completion\_Time}[i,j]) \quad \{1\le i\le N, 1\le j\le M\} \tag{1}$$

As Completion_Time[i, j] is equal to the time that task i is ended on resource j and is calculated to Equation(2):

$$\text{Completion\_Time}[i,j]=\text{Ready}[M]+\text{ETC}[i,j] \tag{2}$$

The purpose of scheduling is that each task should be sent to each resource so that finally Makespan and the number of missed tasks is minimized.

IV. Genetic Algorithm

In 1975, Holland [17] introduced the use of Genetic Algorithm for a optimality process for the first time. Later in 1989, this approach was completed by Goldberg[18]. The invention of this algorithm as an optimization algorithm had been based on Darwinian natural completion and was established based on a strong math theory. This can be applied in the problems which has large searching space, because in this approach unlike single- path approaches, the problem space was searched comprehensively and there will be less possibility for convergence at a local optimized point. Another advantage of this algorithm is that there is no limitation for optimized function such as derived variability, linkage and etc. Also, it can be used in various problems such as linear, nonlinear, continuous, discrete, and it was simply adaptive with various problems.

V. Gravitational Emulation Local Search(GELS)

In 1995, Voudouris and his colleagues [19]suggested GLS algorithm for searching in a searching space and NP-hard solution for the first time. In 2004, Vebster [20] presented it as a strong algorithm and called it GELS algorithm. This algorithm is based on gravitational attraction and it imitates the process of nature for searching within a searching space. Each response has different neighbours which can categorize based on a criteria which is depended on the problem. Obtained neighbours in each group are called neighbours in that dimension. For each dimension, a primary velocity was defined which each dimension has much primary velocity has more appropriate response for problem. GELS algorithm calculated gravitation force within responses in a searching space in two ways. In the first method, a response is selected from local neighbour space of current response and the gravitation force between these two response was calculated. In the second method, the gravitation force among all of the neighbour responses in a neighbour space of current response was calculated and it is not limited to one response. In the movement into searching space, GELS algorithm implements in two methods: the first method is allowed movement from current response to in local neighbour spaces of current response, The second method is allowed movement to the responses out of local neighbour spaces of current response in addition to allowed neighbour responses of current response. Each of these transference methods can be applied with each accounting methods gravitation force, thus, four models are created for GELS algorithm. In 2007[21], Blachandar used of GELS algorithm for solving Traveller Sales Problem (TSP) and compared it with the other algorithms such as Hill-climbing, SA. Conclusions showed that when the size of problem is smaller, all of algorithms are worked similarly, but when the size of problem is larger, GELS algorithm obtained better conclusions comparing to the other algorithms.

5.1. Parameters used in GELS algorithm

(a) Max velocity: Defines the maximum value that any element within the velocity vector can have used to prevent velocities that became too large to use.

(b) Radius: Sets the radius value in the gravitational force formula; used to determine how quickly the gravitational force can increase (or) decrease.

(c) Iterations: Defines a number of iterations of the algorithm that will be allowed to complete before it is automatically terminated (used to ensure that the algorithm will terminate).

(d) Pointer: It is used to identify the direction of movement of the elements in the vectors.

5.2.Gravitational force (GF)

GELS algorithm uses the formula for the gravitational force between the two solutions as

$$F = G(\frac{CU - CA}{R^2})$$

where

G = 6.672 (Universal constant of gravitation)

CU = objective function value of the current solution

CA = objective function value of the candidate solution

R = value of radius parameter

VI. PROPOSED ALGORITHM

The efficiency of Genetic Algorithm is highly depended on the style of chromosomes presentation. Here, a simple method was used to present chromosomes in order to simplify the work of crossover and mutation operators. For encoding the chromosomes natural numbers were used. As the numbers inside the genes are random numbers within 1 to M. Chromosomes length are assumed as task number. Figure 1 shows a simple of the chromosomes style. As it is shown in the figure, for example, task4, t4, is implemented on resource4.

.

T4	T3	T2	T1
4	2	3	1

Fig. 1 Chromosome presentation

Initial Population: Initial population is created randomly. As a source is selected randomly, until the considered task is implemented on it.

Fitness Function: The basic purpose of task scheduling is that the makespan can be minimized. This time equals to total required time until the whole input tasks complete their implementation. It is noted that this time should be always equals or smaller than Max Deadline (MD) among all tasks. In proposed method, a solution is more appropriate for task scheduling problem that in addition to decreasing makespan the number of missed tasks are minimized in it. The other problem which is regarded here is load balancing. Since a solution maybe found that its makespan and the number of its missed tasks are appropriate, but its load balancing on sources isn't appropriate so this factor is considered in fitness function for estimating the solutions. Equation 3 shows the accounting of fitness function.

$$Fitness(ch_i) = \frac{1}{makespan(ch_i)} + \frac{1}{miss\_task * MD} + \frac{1}{LB_i} \quad (3)$$

In this equation, miss_task is the number of tasks which was missing in chromosomei. MD is maximum deadline amount within all tasks. LB_i refers to the CPU load balancing. As the equation shown, however maximum Makespan amount and the number of missed tasks is less, so fitness function is more and it is presented that was being more apparent chromosomes.

Select Operation: Here, tournament operator was used to choose the chromosomes. After accounting the fitness of each chromosomes, a chromosome which has more fitness than the others is selected for applying combination operators on it.

Crossover operation: The crossover operator used in this algorithm is a basic two-point crossover which works as follows: Two random chromosomes were selected by previous phase and exchanges a random part of those.

Combination Operator: Since in mutation operator, the points which the changes are applied on this are chosen randomly, so in proposed method instead of this operator, chromosomes changes were done through GELS algorithm for making the neighbour solutions. Because the changing ways in this algorithm isn't mere randomly. When each chromosome was selected by previous phase, was controlled by GELS algorithm until the solution of its neighbour was obtained. Each gene of chromosome is considered as a dimension of the problem. In fact the problem's dimensions are just the neighbouring solutions which are obtained by changing the current solution. Initial velocity is given to each of the

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:5, No:7, 2011

problem's dimensions that is done randomly and ranges from 1 to the maximum of the initial velocity. A gene which have more velocity was selected and its values was changed randomly with the number 1 to M. After that, fitness amount is calculated using Equation 3. If candidate solution is improved comparing to current solution, it is substituted to its parent chromosomes in new population. Nevertheless, it is not copied in new population.Then, the amount of the force between candidate solution and current solution is calculated using Equation 4 and its amount is added to primary velocity vector related to the dimension which the candidate response is obtained from it until its primary velocity vector is updated.

$$F = G * \frac{Fitness(candidate\_ch_i) - Fitness(current\_ch_i)}{R^2} \quad (4)$$

Finalizing Conditions: Algorithm is finished when the primary velocity equal to zero for all dimensions or the number of algorithm iteration received to its maximum number. Figure 2 shows GA-GELS pseudo code.

1. Initial the K number of chromosome
2. Evaluates all individuals in the population using equation(4)
3. Dimensioning (n)
4. Velocity_Vector[1..n]=Initial velocity for each Dimension()
5. While(i<=max_iteration or Velocity_Vector[...]≠0){
6. /* select K/2 chromosomes using the tournament selection from K chromosomes*/
7. Crossover the selected chromosomes
8. Direction=max(Velocity_Vector[…])
9. Selected chromosomes set as current_chrom and make candidate _chrom with direction
10. If(Fitness(candidate chrom)> Fitness(current chrom))

Fig. 2 GA-GELS pseudo code

VII. Implementation and Experiments

The GA-GELS is implemented using java software on hardware with these characteristics: CPU 2.66GHZ, 4GB RAM, Win XP operating system. In proposed algorithm has been assumed PC=0.98, PM= 0.05. The amounts of primary velocity vector for chromosome was assumed among 1 to the whole number of tasks, randomly. Conclusions of Simulation of comparison among proposed algorithm (GA-GELS) and GELS, Genetic, GSA algorithm were shown in figures 3, 4.
In Figure 3, a diagram is shown in which the number of tasks was scheduled between 20 and 60 on 20 resources using considered algorithms. As it is shown, if the number of tasks increased, Makespan is increased too. Within the scheduled algorithm it is showed that the proposed algorithm produces less Makespan comparing to the other.

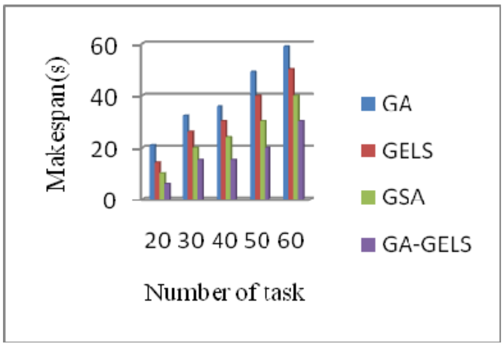

Fig. 3 Comparison of Makespan

In Figure 4, the algorithms are compared together from missing rate of tasks view. In designed diagram, the fitness amount is against to the rate of missed tasks. Since the diagram is shown, whatever fitness amount is increased, the rate of missed tasks is decreased. It means that the number of missed tasks are decreased as a result of completion of their Makespan. The figure shows that GA-GELS has less missed tasks comparing to above mentioned algorithms.

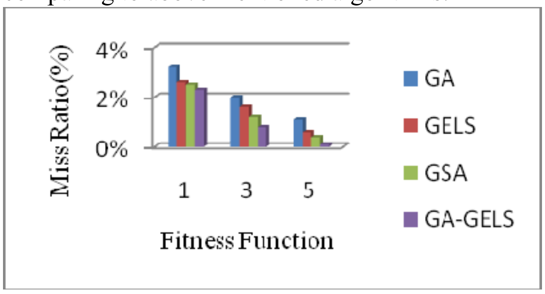

Fig. 4 Comparison the average miss ratio of different fitness functions algorithms

VIII. Conclusion

This paper presents a novel task scheduling method based on hybrid Genetic and GELS(GA-GELS) algorithm to solve grid scheduling problem to minimize makespan and missed task. Each chromosome represents a feasible solution. Local searching algorithms such as hill-climbing, SA,...always move to the solutions with better amount of purpose function and they search the problem space randomly. Although GELS algorithm have special behavior of greedy algorithms, but it doesn't always move to a solution with better amount of purpose function directly, but it works by examining existed solutions. Although, GELS algorithm have some random elements in itself but it doesn't always move within them in the same way that is why it doesn't stop on local optimal solution. By combining GELS algorithm advantages with genetic, convergence velocity and finding optimum response in genetic were improved. The proposed algorithm was compared to the other algorithms and the simulation conclusions showed that the GA-GELS produced less Makespan comparing the other algorithms and also it decreases the number of missed tasks.

.

REFERENCES

[1] Kołodzie. J, Xhafa. F, “Meeting security and user behavior requirements in Grid scheduling ", Simulation Modeling Practice and Theory Vol.19,Iss. 1, pp. 213–226, 2011.DOI: http://dx.doi.org/10.1016/j.simpat.2010.06.007

[2] Tseng. L. Y, Chin. Y. H, Wang. S. C, “The anatomy study of high performance task scheduling algorithm for Grid computing system", Computer Standards & Interfaces Vol. 31, Iss. 4, pp. 713–722, 2009. DOI: http://dx.doi.org/10.1016/j.csi.2008.09.017

[3] M. Shojafar, S. Barzegar, M. R. Meybodi, “A new Method on Resource Scheduling in grid systems based on Hierarchical Stochastic Petri net”, First International Conference on Information, Networking and Automation (ICINA 2010), Vol. 9, No 2,pp. V9-175-180, China, 2010. DOI: http://www.irexpert.ir/UploadedFiles/Records/16732_201008142030093424zmd4dhsos.changeto(pdf)

[4] Shenassa. M. H, Mahmoodi. M, "A novel intelligent method for task scheduling in multiprocessor systems using genetic algorithm", journal of Franklin Institute, Elsevier, Vol.343, Iss. 4-5, pp. 361-371, 2006. DOI: http://dx.doi.org/10.1016/j.jfranklin.2006.02.022

[5] Pourhaji Kazem A. A, Rahmani A. M. and Habibi Aghdam. H, “A Modified Simulated Annealing Algorithm for Static Scheduling in Grid Computing”, International Conference on Computer Science and Information Technology 2008 (ICCSIT 2008), Singapore August 29 – September, pp. 623-627, 2008. DOI: http://dx.doi.org/10.1109/ICCSIT.2008.163

[6] Benedict. SH, Vasudevan. V, “Improving scheduling of scientific workflows using tabu search for computational grids”, Information Technology Journal Vol.7, No. 1, pp. 91- 97, 2008. DOI: http://scialert.net/abstract/?doi=itj.2008.91.97

[7] Rahmani. A. M, Rezvani. M, "A Novel Genetic Algorithm for Static Task Scheduling in Distributed Systems", International Journal of Computer Theory and Engineering, Vol. 1, No. 1, pp. 1793- 8201, April 2009. DOI: http://www.etlibrary.org/?m=fbook&a=down&aid=10

[8] Abdulal. W, Jadaan. O. A, Jabas. A, Ramachandram. S, "An Improved Rank-based Genetic Algorithm with Limited Iterations for Grid Scheduling", IEEE Symposium on Industrial Electronics and Applications, pp. 215-220, October 2009. DOI: http://10.1109/ISIEA.2009.5356468

[9] Tamilarasi. A, Anantha kumar. T, "An enhanced genetic algorithm with simulated annealing for job-shop scheduling", International Journal of Engineering, Science and Technology, Vol. 2, No. 1, pp. 144- 151, 2010. DOI: http://www.doaj.org/doaj?func=openurl&genre=article&issn=21412820&date=2010&volume=2&issue=1&spage=144

[10] Omaraa. F. A, Arafa. M. M,” Genetic algorithms for task scheduling problem”, Journal Parallel Distributed Computing, Vol. 70, Iss. 1, pp. 13-22, 2010.DOI: http://dx.doi.org/10.1016/j.jpdc.2009.09.009

[11] Khanli. L. M, Etminan Far, Ghaffari. A, “Reliable Job Scheduler using RFOH in Grid Computing”, Journal of Emerging Trends in Computing and Information Sciences, Vol. 1, No. 1, pp. 43- 47, July 2010. DOI: DOI: http://www.doaj.org/doaj?func=openurl&genre=article&issn=20798407&date=2010&volume=1&issue=1&spage=43

[12] Tao. Q, Chang. H, Yi. Y, Gu. CH,”A Grid Workflow Scheduling Optimization Approach for e-Business Application”, International Conference on E-Business and E-Government, pp. 168- 171, 2010. DOI: http://dx.doi.org/10.1109/ICEE.2010.50

[13] Gharooni fard. G, Moein darbari. F, Deldari. H, Morvaridi. A, "Scheduling of scientific workflows using a chaos- genetic algorithm", International Conference on Computational Science ICCS, pp. 1439-1448, 2010. DOI: http://dx.doi.org/10.1016/j.procs.2010.04.160

[14] Lifeng Ai and Maolin Tang, “QoS-Based Web Service Composition Accommodating Inter-Service Dependencies Using Minimal-Conflict Hill-Climbing Repair Genetic Algorithm”, Fourth IEEE International Conference on Science, pp. 119- 126, 2008.DOI: http://dx.doi.org/10.1109/eScience.2008.110

[15] Barzegar. B, Rahmani. A. M, Zamani far. K, “Gravitational Emulation Local Search Algorithm for Advanced Reservation and Scheduling in Grid Systems”, First Asian Himalayas International Conference on (2009), pp. 1-5, 2009. DOI: http://dx.doi.org/10.1109/AHICI.2009.5340301

[16] Braun. T. D, Siegel. H. J,Beck. N, Boloni. L. L, Maheswaran. M, Reuther. A. L, Robertson. J. P, Theys. M. D, Yao. B, Hensgen. D, Freund. R. F,”A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems”, Journal of Parallel and distributed Computing Vol. 61, No. 6, pp. 680- 1983, 2001.DOI: http://dx.doi.org/10.1006/jpdc.2000.1714

[17] Holland. J, “Adaptation in Natural and Artificial Systems”, University of Michigan Press, Ann Arbor, ISBN: 0-262-58111-6, pp. 228, 1975.DOI: http://www.amazon.com/Adaptation-Natural-Artificial-Systems-Introductory/dp/0262581116

[18] Goldberg. D.E, “Genetic Algorithms in Search, Optimization and Machine Learning”, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, ISBN: 0201157675, pp. 432, 1989. DOI: http://www.amazon.com/Genetic-Algorithms-Optimization-Machine-Learning/dp/0201157675

[19] Voudouris, chris, Edward Tsang, Guided Local Search. Technical Report CSM-247, Department of Computer Science, University of Essex, UK, August 1995.

[20] Barry Lynn Webster, “Solving Combinatorial Optimization Problems Using a New Algorithm Based on Gravitational Attraction”, Ph.D. Thesis, Florida Institute of Technology Melbourne, FL, USA, May 2004.DOI: http://dx.doi.org/10.1109/T-C.1973.223690

[21] Raja Balachandar. S, Kannan. K, “Randomized gravitational emulation search algorithm for symmetric traveling salesman problem”, Applied Mathematics and Computation, Vol. 192, Iss. 2, pp. 413–421, 2007.DOI: http://dx.doi.org/10.1016/j.amc.2007.03.019