International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:3, No:11, 2009

A Meta-Heuristic algorithm for Vertex covering problem Based on Gravity

S. Raja Balachandar and K.Kannan

Abstract—A new Meta heuristic approach called ”Randomized gravitational emulation search algorithm (RGES)” for solving vertex covering problems has been designed. This algorithm is found upon introducing randomization concept along with the two of the four primary parameters ’velocity’ and ’gravity’ in physics. A new heuristic operator is introduced in the domain of RGES to maintain feasibility specifically for the vertex covering problem to yield best solutions. The performance of this algorithm has been evaluated on a large set of benchmark problems from OR-library. Computational results showed that the randomized gravitational emulation search algorithm - based heuristic is capable of producing high quality solutions. The performance of this heuristic when compared with other existing heuristic algorithms is found to be excellent in terms of solution quality.

Keywords—Vertex covering Problem, Velocity, Gravitational Force, Newton’s Law, Meta Heuristic, Combinatorial optimization.

I. Introduction

There is a class of problems, whose exponential complexities have been established theoretically are known as NP problems. Designing polynomial time algorithms for such a class of problems is still open. Due to the demand for solving such problems, Researchers are constantly attempting to provide heuristic solutions one after the other focusing the optimality by introducing several operators with salient features such as (i) reducing the computational complexity, (ii) randomization etc.,
Some NP problems are Set covering problem, Traveling salesman problem, Problem of Hamiltonian paths, Knapsack problem, Problem of optimal graph coloring. If a polynomial time solution can be found for any of these problems, then all of the NP problems would have polynomial solutions. NP complete problems are described more detail in [8]. In 1972, in a landmark paper Karp[19] has shown that the vertex cover problem is NP - complete, meaning that it is exceedingly unlikely that to find an algorithm with polynomial worst - case running time. The minimum vertex cover problem remains NP - complete even for certain restricted graphs, for example, the bounded degree graphs[9] .

Vertex cover problem (VCP) has attracted researchers and practitioners not only because of the NP - completeness but also because of many difficult real - life problems which can be formulated as instances of the minimum weighted vertex cover. Examples of such areas where the minimum weighted vertex cover problem occurs in real world applications are communications, particularly in wireless telecommunications, civil, electrical engineering, circuit design, network flow, problem of placing guards[32] are worthmentioning. Though both exact (optimal) and heuristic approaches have been presented in the literature, this problem is still a difficult NP-complete problem. A vertex cover for an undirected graph G = (V, E) is set of vertices such that all the edges in the graph are incident upon at least one vertex in the cover. The minimum cardinality vertex cover for a graph is a vertex cover with the least number of vertices. A weighted vertex cover problem(WVCP) is defined as follows: Given G(V, E) and weight function $w : V \to R$, find a cover of minimum total weight. Thus the problem can be mathematically transformed into the following optimization problem

minimize

$$\sum_{j=1}^{n} w_j v_j \tag{1}$$

subject to

$$\sum_{j=1}^{n} v_i + v_j \geq 1, \forall (v_i, v_j) \in E \tag{2}$$

$$v_j \in \{0, 1\}, j = 1, 2, 3, ..., n \tag{3}$$

Equation (2) ensures that each edge is covered by at least one vertex and (3) is the integral of constraint. The cost coefficients w_j are equal to 1 the problem is referred to as the unicost VCP, otherwise,the problem is called the weighted or weighted VCP.

The minimum weighted vertex cover problem is closely related to many other hard probelms and it is of interest to the researchers in the field of design of optimization and approximation algorithms. Minimum weighted vertex cover problem is a special case of set covering problem[5][12] [14]and the independent set problem[2][9][19] is similar to the minimum vertex cover problem because a minimum vertex cover defines a maximum independent set and vice versa. Another interesting problem is closely related to the minimum vertex is the edge cover which seeks the smallest set of edges such that each vertex is included in one of the edges.

In this paper, a new optimization algorithm based on the law of gravity, namely Randomized gravitational emulation search algorithm (RGES) is proposed. This algorithm is based on the Newtonian gravity: ” Every particle in the universe attracts every other particle with a force that is directly proportional to the product of their masses and inversely proportional to

S.Raja Balachandar is with the Department of Mathematics, SASTRA University,Thanjavur,INDIA, e-mail: srbala@maths.sastra.edu

K.Kannan is with the Department of Mathematics, SASTRA University,Thanjavur,INDIA, e-mail: kkannan@maths.sastra.edu

the square of the distance between them".
This article demonstrates that RGES technique is capable of producing better quality results for the large size set covering problem than other heuristic approaches.
This paper is organized as follows: A brief survey of various approaches pertaining to this problem is elucidated in section II. In section III, we introduce the basic concepts of our algorithm. The proposed RGES is presented in section IV. The algorithm's utility is illustrated with help of benchmark problems in section V and we include the extensive comparative study of result of our heuristic with existing state-of-art heuristics. Salient features of this algorithm are enumerated in section VI, finally concluding remarks are given in section VII.

II. Previous Work

WVCP is known to be NP-Hard, even if all the weights are 1 and the graph is planar[8] . Due to computational intractability of the MWVC problem, many researchers have instead focused their attention on the design of heuristic/approximation algorithm for delivering quality solutions in a reasonable time.

Johnson[16] gave the first(greedy) logarithmic ratio approximation for the unweighted uncapacited cover problem. Consider the case where all vertices have the same weight. Since goal becomes the minimization of the cardinality of a subset of V such that each edge (u,v) in E, at least one of u and v is in the subset, it is intuitive to successively select the vertex with the largest degree until all of the edges are covered by the vertices in subset of V. This straightforward heuristic can be further generalized as and applied to MWVCP. The generalization proposed and analyzed by Chvatal[3] collects a vertex at each stage with the smallest ratio between its weight and current degree. Clarkson[4] presented a heuristic algorithm that exhibits a performance guarantee of 2.

Pitt[27] gave a randomized algorithm which randomly selects an end vertex of an arbitrary edge with a probability inversely proportional to its weight. For a comprehensive survey on the analysis of approximation algorithms for MWVC, the reader is referred to Monien and Speckenmeyer[30] , Motwani[25] , Hastad[13] , Shyu, Yin and Lin[31] , Likas and Stafylopatis [22] . The first fixed parameter tractable algorithm for k - vertex cover problem was done by Fellows [7]. Recently, Dehne et all[6] have reported that they used fixed parameter tractable algorithm to solve the minimum vertex cover problem on coarse-grained parallel machines successfully. Neidermeier and Rossmanith[26] presented efficient fixed parameter algorithm for the minimum weighted vertex cover problem. Shyu[31] presented a meta-heuristic approach Ant colony Optimization Algorithm(ACO) for WVCP and compared the performance of ACO with other heuristic and meta-heuristic like, genetic algorithm, tabu search, and simulated annealing for random graphs.

In this paper, we have designed a meta heuristic algorithm based on gravity and we enhanced the performance of RGES through feasibility operator to obtain best solutions at less computational cost.

III. The law of gravity

The gravitation is the tendency of masses to accelerate toward each other. It is one of the four fundamental interactions in nature [29] (the others are: the electromagnetic force, the weak nuclear force, and the strong nuclear force). Every particle in the universe attracts every other particle. Gravity is everywhere. The inescapability of gravity makes it different from all other natural forces. The way Newton's gravitational force behaves is called "action at a distance". This means gravity acts between separated particles without any intermediary and without any delay. In the Newton law of gravity, each particle attracts every other particle with a 'gravitational force' [29][15] [28]. The gravitational force between two particles is directly proportional to the product of their masses and inversely proportional to the square of the distance between them [15]:

$$F = \frac{GM_1M_2}{R^2} \tag{4}$$

where F is the magnitude of the gravitational force, G is gravitational constant, M_1 and M_2 are the mass of the first and second particles respectively, and R is the distance between the two particles. Newton's second law says that when a force, F, is applied to a particle, its acceleration, a, depends only on the force and its mass, M [15]:

$$a = \frac{F}{M} \tag{5}$$

Based on (4) and (5), there is an attracting gravity force among all particles of the universe where the effect of bigger and the closer particle is higher. An increase in the distance between two particles means decreasing the gravity force between them as it is illustrated in Fig.1. In this figure, F_{1j} is the force that acting on M_1 from M_j and F_1 is the overall force that acts on M_1 and causes the acceleration vector a_1. In addition, due to the effect of decreasing gravity, the actual value of the "'gravitational constant" depends on the actual age of the universe. Eq. (6) gives the decrease of the gravitational constant, G, with the age [23]:

$$G(t) = G(t_o) \times (\frac{t_o}{t})^{\beta}, \beta < 1, \tag{6}$$

where G(t) is the value of the gravitational constant at time t. $G(t_o)$ is the value of the gravitational constant at the first cosmic quantum-interval of time t_o [23]. Three kinds of masses are defined in theoretical physics:
Active gravitational mass, M_a, is a measure of the strength of the gravitational field due to a particular object. Gravitational field of an object with small active gravitational mass is weaker than the object with more active gravitational mass.
Passive gravitational mass, M_p, is a measure of the strength of an object's interaction with the gravitational field. Within the same gravitational field, an object with a smaller passive gravitational mass experiences a smaller force than an object with a larger passive gravitational mass.
Inertial mass, M_i, is a measure of an object resistance to changing its state of motion when a force is applied. An object with large inertial mass changes its motion more slowly, and an object with small inertial mass changes it rapidly.

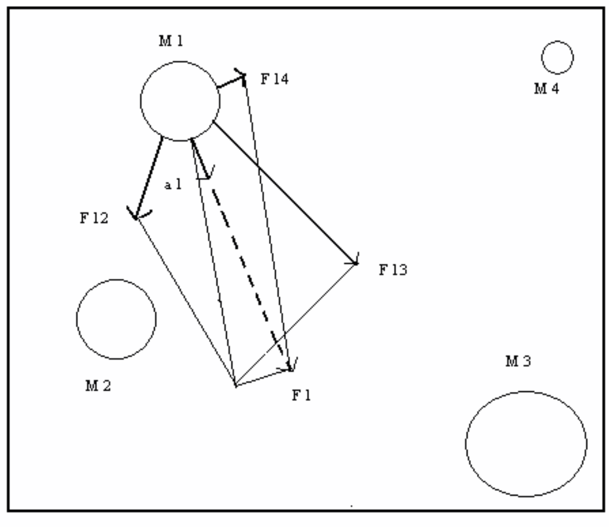

Fig. 1. Every mass accelerate toward the result force that act it from the other masses

Now, considering the above-mentioned aspects, we rewrite Newton's laws. The gravitational force, F_{ij}, that acts on mass i by mass j, is proportional to the product of the active gravitational of mass j and passive gravitational of mass i, and inversely proportional to the square distance between them. a_i is proportional to F_{ij} and inversely proportional to inertia mass of i. More precisely, one can rewrite Eqs. (4) and (5) as follows:

$$F_{ij} = \frac{GM_{aj}M_{pi}}{R^2}, \tag{7}$$

$$a_i = \frac{F_{ij}}{M_{ii}}, \tag{8}$$

where M_{aj} and M_{pi} represent the active gravitational mass of particle i and passive gravitational mass of particle j, respectively, and M_{ii} represents the inertia mass of particle i.

Although inertial mass, passive gravitational mass, and active gravitational mass are conceptually distinct, no experiment has ever unambiguously demonstrated any difference between them. The theory of general relativity rests on the assumption that inertial and passive gravitational mass are equivalent. This is known as the weak equivalence principle [20] [23]. Standard general relativity also assumes the equivalence of inertial mass and active gravitational mass; this equivalence is sometimes called the strong equivalent principle [20].

IV. Randomized Gravitational Emulation Search Algorithm(RGES)

In this section, we introduce our optimization algorithm based on the law of gravity [28]. In the proposed algorithm, agents are considered as objects and their performance is measured by their masses. All these objects attract each other by the gravity force, and this force causes a global movement of all objects towards the objects with heavier masses. Hence, masses cooperate using a direct form of communication, through gravitational force. The heavy masses - which correspond to good solutions - move more slowly than lighter ones, this guarantees the exploitation step of the algorithm. In RGES, each mass (agent) has four specifications: position, inertial mass, active gravitational mass, and passive gravitational mass. The position of the mass corresponds to a solution of the problem, and its gravitational and inertial masses are determined using a fitness function. In other words, each mass presents a solution, and the algorithm is navigated by properly adjusting the gravitational and inertia masses. By lapse of time, we expect that masses be attracted by the heaviest mass. This mass will present an optimum solution in the search space. The RGES could be considered as an isolated system of masses. It is like a small artificial world of masses obeying the Newtonian laws of gravitation and motion. More precisely, masses obey the following laws:

Law of gravity: each particle attracts every other particle and the gravitational force between two particles is directly proportional to the product of their masses and inversely proportional to the distance between them, R. We use here R instead of R^2, because according to our experiment results, R provides better results than R^2 in all experimental cases.

Law of motion: the current velocity of any mass is equal to the sum of the fraction of its previous velocity and the variation in the velocity. Variation in the velocity or acceleration of any mass is equal to the force acted on the system divided by mass of inertia.

A. Initiation

Now, consider a system with N agents (masses). We define the position of the ith agent by:

$$X_i = (x_i^1, x_i^2, ..., x_i^d, ..., x_i^n) \quad for\ i = 1, 2, 3, ..., N, \tag{9}$$

where x_i^d presents the position of ith agent in the dth dimension. At a specific time 't', we define the force acting on mass 'i' from mass 'j' as following:

$$F_{ij}^d(t) = G(t)\frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \in}(x_i^d(t) - x_i^d(t)), \tag{10}$$

where M_{aj} is the active gravitational mass related to agent j, M_{pi} is the passive gravitational mass related to agent i, G(t) is gravitational constant at time t, $\in$ is a small constant, and $R_{ij}(t)$ is the Euclidian distance between two agents i and j:

$$R_{ij} = \|X_i(t), X_j(t)\|_2\,, \tag{11}$$

To give a stochastic characteristic to our algorithm, we suppose that the total force that acts on agent i in a dimension d be a randomly weighted sum of dth components of the forces exerted from other agents:

$$F_i^d(t) = \sum_{j=1, j\neq i}^{N} rand_j F_{ij}^d(t), \tag{12}$$

where $rand_j$ is a random number in the interval [0, 1]. Hence, by the law of motion, the acceleration of the agent i at time t, and in direction dth, $a_i^d(t)$,is given as follows:

$$a_i^d(t) = \frac{F_i^d(t)}{M_i i(t)}, \tag{13}$$

where M_{ii} is the inertial mass of ith agent. Furthermore, the next velocity of an agent is considered as a fraction of its current velocity added to its acceleration. Therefore, its position and its velocity could be calculated as follows:

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t), \tag{14}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1), \tag{15}$$

where $rand_i$ is a uniform random variable in the interval [0, 1]. We use this random number to give a randomized characteristic to the search. The gravitational constant, G, is initialized at the beginning and will be reduced with time to control the search accuracy. In other words, G is a function of the initial value (G_o) and time (t):

$$G(t) = G(G_o, t), \tag{16}$$

B. Evaluation of fitness and updating

Gravitational and inertia masses are simply calculated by the fitness evaluation. A heavier mass means a more efficient agent. This means that better agents have higher attractions and walk more slowly. Assuming the equality of the gravitational and inertia mass, the values of masses are calculated using the map of fitness. We update the gravitational and inertial masses by the following equations:

$$M_{ai} = M_{pi} = M_{ii} = M_i, i = 1, 2, 3, ..., N, \tag{17}$$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \tag{18}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} \ m_j(t)}, \tag{19}$$

where $fit_i(t)$ represent the fitness value of the agent i at time t, and, worst(t) and best(t) are defined as follows for a minimization problem:

$$best(t) = \underbrace{min}_{j \in 1,...,N} \ fit_j(t), \tag{20}$$

$$worst(t) = \underbrace{max}_{j \in 1,...,N} \ fit_j(t), \tag{21}$$

One way to perform a good compromise between exploration and exploitation is to reduce the number of agents with lapse of time in Eq. (12). Hence, we propose only a set of agents with bigger mass apply their force to the other. However, we should be careful of using this policy because it may reduce the exploration power and increase the exploitation capability. We remind that in order to avoid trapping in a local optimum the algorithm must use the exploration at beginning. By lapse of iterations, exploration must fade out and exploitation must fade in. To improve the performance of RGES by controlling exploration and exploitation only the Kbest agents will attract the others. Kbest is a function of time, with the initial value K_o at the beginning and decreasing with time. In such a way, at the beginning, all agents apply the force, and as time passes, Kbest is decreased linearly and at the end there will be just one agent applying force to the others. Therefore, Eq.(12) could be modified as:

$$F_i^d(t) = \sum_{j \in Kbest, j \neq i} rand_j F_{ij}^d(t), \tag{22}$$

where Kbest is the set of first K agents with the best fitness value and biggest mass.

C. Repair operator

The solutions(agents) may violate constraints. To make all the solutions feasible an additional operator is needed. Here a proposed heuristic operator consists of two phases namely ADD phase and DROP phase that maintains the feasibility of the solutions in the neighborhood being generated. The steps required to make each solution feasible involve the identification of all uncovered rows and the addition of columns such that all rows are covered. This is done by the ADD phase. Once columns are added, a solution becomes feasible. DROP phase (a local optimization procedure) is applied to remove any redundant column such that by removing it from the solution, the solution still remains feasible. In the algorithm, steps (i) and (ii) identify the uncovered rows and add the least cost column to the solution vector. Steps (iii) and (iv) identify the redundant column with high cost and dropped from the solution. The time complexity of this repair operator is O(mn).
Different steps of the repair operator are the followings

S_{1xn} = solution vector
B_{nxm} = transpose of the adjacency matrix
D_{1xn} = temporary solution vector
C_{1xm} = counter vector (0 entry of any position is used to identify the uncovered rows)
(i)C = $S \times B$ (matrix multiplication)
(ii) ADD Phase
(a) For each 0 entry in C , find the first column j(cost of j is in increasing order)
(b) Add j to S ie., S(j) = 1.
(c) D = S (temporary)
(iii)DROP Phase
(a) Identify the column j (cost in the decreasing order)
(b) Remove j from D, if $C = D \times B$ have no zero entry, ie., D(j) = 0.
(c) S=D is a feasible solution for SCP that contains no redundant columns.

The different steps of the proposed RGES algorithm are the followings:

(a) Search space identification.
(b) Randomized initialization.
(c) Repair operator.
(d) Fitness evaluation of agents.

(e) Update G(t), best(t), worst(t) and $M_i(t)$ for $i = 1, 2, ..., N$.
(f) Calculation of the total force in different directions.
(g) Calculation of acceleration and velocity.
(h) Updating agents' position.
(i) Repeat steps c to h until the stop criteria is reached.
(j) End.

V. Experimental results and analysis

The RGES has been coded in MATLAB7. The heuristic tested on 44 instances corresponding to two groups; the first group contains the test instances namely mvcp1-12 with weight 1, it is tested by Khuri [21] and second group consists of 32 test instances namely wvcp1-30 with weight which are randomly generated [1] [33]and used by shyu[31].

In our experimental study, 10 trials RGES heuristic were made for each of the test problems with n(number of vertices) random solutions. Each trial was terminated, once 1000 iterations are completed or velocity is equal to zero. This algorithm was implemented in C and tested in P-IV, 3.2GHz processor and 512 MB RAM running under Windows XP.

Experiment 1: Minimum Vertex Cover

In order to bring out the efficiency of the proposed RGES algorithm the solutions of the same set of test instances have been compared with the other approaches (Tabu Search, Genetic Algorithm, Simulated Annealing). TABLE I provides a summary of the solutions obtained by these methods and solution quality for these different heuristics namely average gap (average = (solution - BKS)/BKS x 100), number of optimum solutions and best solutions. RGES found the optimal / best-known solutions for all the 12 test instances. From this table, we can observe that RGES, CFT, and Meta-RaPS have zero deviation from the best-known or optimal solutions for these test problems.

TABLE I
Solutions of minimum vertex cover problem

Instance	Opt Obj Value	GA	SA	TS	RGES
100-01	53	54	55	55	53
100-02	50	53	53	54	50
100-03	55	57	57	55	55
100-04	54	55	55	54	54
100-05	55	55	57	57	55
PS100	34	34	34	34	34
200-01	110	113	113	131	110
200-02	110	120	120	132	110
200-03	110	128	120	130	110
200-04	110	140	136	140	110
200-05	110	110	110	110	110
PS202	68	68	68	68	68
Avg Error	-	5.74	4,56	8.33	0.00

Experiment 2: Weighted Minimum Vertex Cover In this experiment the parameter set opted like small - large scale problems , the number of vertices n is 50,100,150,200,250 or 300. For each setting of n, we let m be ranged from 50 to 5000. For practical considerations, we assume that the wight on each vertex is propotional to the degree of the vertex (more transportation benefits)on a vertex might induce more weight (more running costs) on it. Let weight w(i) on vertex i be randomly distruted over the interval [1, d(i)^2], where d(i) is the degree of the vertex i, 1 and ten randomly generated data instances for each pair of n an m.

We implemented the following heuristics

REP(pite gr,1985)[27]. The method randomly selects an end vertex of an arbitrary edge considering the probability inversely proportional to its weight.
GM (Chavatal, 1979; Motwani,1992)[3][24]. The method greedily selects the vertex with minimum ratio between its weight and current degree.
MGM(Clarkson,1983)[4]. The method greedily selects the vertex with minimum ratio between its weight and current degree where the weight is modified as the heuristic progesses.
ACO(Shyu, 2004)[31]. Ant Colony Optimization algorithm.
SA(johnson)[17][18]. Simulated Annealing.
TS(glover)[10][11]. Tabu Search.
GA(Khuri, 1994)[21]. Genetic Algorithm.

The results of RGES,REP,GM,MGM,ACO,TS,GA,SA for second set are presented in TABLE II. The first two columns indicate that number of vertices (V) and number of edges (E). The next SEVEN columns indicated that best solutions of differnt algorithms. It is clear that RGES found minimum solutions for all the 32 test instances. So the RGES has identified high quality solutions for large instances also.Note that in the point of solution quality, GM,SA and TS give deviations of 7.98, 6.50 and 6.53 percentage from RGES, respectively. Since the weights on vertices in this experiment were randomly generated over the interval [1, d(i)^2],a vertex with larger degree would be cline to having a heavier weight. The heuristic that GM deployed would be less competitive in such a situation as compared to ACO and REGS, even when GM was further improved by simulated annealing or tabu search. Hence, we can see that the quality of the solution delivered by RGES are much better than the other heuristics involved in this experiment even though weights on vertexes are proportional to the degrees.

VI. Features of Algorithm

To see how the proposed algorithm is efficient some remarks are noted: Since each agent could observe the performance of the others, the gravitational force is an information-transferring tool. Due to the force that acts on an agent from its neighborhood agents, it can see space around itself. A heavy mass has a large effective attraction radius and hence a great intensity of attraction. Therefore, agents with a higher performance have a greater gravitational mass. As a result, the agents tend to move toward the best agent. The inertia mass is against the motion and make the mass movement slow. Hence, agents with heavy inertia mass move slowly and hence search the space more locally. So, it can be considered as an adaptive learning rate. Gravitational constant adjusts the accuracy of the search, so it decreases with time (similar to the temperature in a Simulated Annealing algorithm). RGES is a memory-less algorithm. However, it works efficiently like the algorithms with memory.

Our experimental results show the good convergence rate of the RGES. Here, we assume that the gravitational and the inertia masses are the same. However, for some applications different values for them can be used. A bigger inertia mass provides a slower motion of agents in the search space and hence a more precise search. Conversely, a bigger gravitational mass causes a higher attraction of agents. This permits a faster convergence.

VII. Conclusion

A feasibility operator based heuristic for the vertex covering problem based on RGES has been developed. Randomization enables the algorithm to escape from the local search and pave a way leading to find optimal solutions. Computational results indicate that our heuristic is able to generate optimal solutions for small size problems in less time. For large size problems the deviation from the optimal solutions are very less and are much below the deviations obtained by other existing algorithms. The successful applications of the RGES approach to complex optimization problems are extending the study of meta heuristic. For further research, it is of potential interest to apply the RGES approach to other problems that are not necessarily based on graphs. Extending the generic RGES model by incorporating specific behaviours of physical parameters or computer technologies, such as parallel processing, to enhance its problem solving capability may be another research direction.

TABLE II
Solutions of Minimum Weighted vertex cover problem

N	M	REP	GM	MGM	SA	TS	ACO	RGES
50	50	113.1	95.1	98.7	93.5	93.5	83.9	83.9
	100	355	305.2	312.7	299.9	299.9	276.2	276.2
	250	2319	2051.5	2138.4	1990.9	2006.7	1886.8	1886.4
	500	9189.4	8196.6	8635.6	8115.5	8115.5	7915.9	7914.5
	750	22246.7	20604.9	21676	20574.6	20604.9	20134.1	20134.1
100	50	90.3	73.2	73.8	71.7	71.7	67.4	67.4
	100	224.9	186.1	198.9	183.7	184.1	169.1	169.1
	250	1150.2	995.5	1053.9	986.7	983.9	901.7	890.4
	500	4740.8	3991.8	4307.7	3937.8	3937.8	3726.7	3725.3
	750	10236.1	9256.9	9771.9	9172.4	9172.4	8754.5	8745.5
150	50	88.6	71.2	72.6	70.6	70.6	65.8	65.8
	100	196.5	159.6	165.3	157.9	157.9	144.7	144.7
	250	848.5	692.5	735.1	679.5	679.1	625.7	624.4
	500	3148.9	2577.6	2734.9	2519.4	2526.2	2375	2365.2
	750	7441	6236.1	6628.7	6090.8	6105.9	5799.2	5798.6
200	50	74.3	62	63.2	61.2	61.2	59.6	59.6
	100	173.5	146.8	151	145.1	145.2	134.7	132.6
	250	658.2	543.2	576.4	537	537	488.7	488.4
	500	2368.3	2004.6	2157.8	1989	1989	1843.6	1843.6
	750	5165.7	4422.9	4727.4	4376.4	4383.7	4112.8	4112.8
250	250	602.7	469.1	492.2	462.1	463.1	423.2	423.2
	500	1933.5	1602.6	1697.9	1591.8	1591.8	1457.4	1457.4
	750	4332.2	3564.1	3888.6	3512.1	3513.3	3315.9	3315.9
	1000	7723.3	6554.6	6954.4	6438.7	6436	6058.2	6058.2
	2000	31475.8	27360.2	29130.6	26925.4	26864	26149.1	26149.1
	5000	193232.3	176245.2	183612.8	174037.5	173902.9	171917.2	171917.2
300	250	534	447.3	469.9	441.2	441.3	403.9	403.9
	500	1648.1	1361.3	1451.1	1348.6	1347.4	1239.1	1239.1
	750	3596	2924.2	3121.2	2878.7	2879.4	2678.2	2678.2
	1000	6428	5274.3	5718.4	5229.9	5227.4	4895.5	4895.5
	2000	26106.6	22432.5	23997.8	22061.2	21983.6	21295.2	21295.2
	5000	163003.3	147406.4	154929.6	145276.6	145121.4	143243.5	143243.5

REFERENCES

[1] Bollobas. B : Random graphs (2nd Ed.). Cambridge, UK: Cambridge University press, 2001.

[2] Berman. P and Fujito. T: On approximation properties of the independent set problem for low degree graphs, Theory of Computing Syst., vol. 32, pp. 115 - 132, 1999.

[3] Chvatal, V. (1979). ”A Greedy-Heuristic for the Set Cover Problem.”Mathematics of Operations Research 4, 233-235.

[4] Clarkson, K.L (1983). ”A Modification of the Greedy Algorithm for Vertex Cover.” Information Processing Lettters 16, 23-25.

[5] Cormen. T. H, C. E. Leiserson, R. L. R., and Stein. C: Introduction to algorithms, 2nd ed., McGraw - Hill, New York , 2001.

[6] Dehne. F, et al.: Solving large FPT problems on coarse grained parallel machines, Available: http://www.scs.carleton.ca/fpt/papers/index.htm.

[7] Fellows. M. R: On the complexity of vertex cover problems, Technical report, Computer science department, University of New Mexico, 1988.

[8] Garey. M. R, Johnson. D. S: Computers and Intractability: A Guide to the theory NP - completeness. San Francisco: Freeman ,1979.

[9] Garey. M. R, Johnson. D. S, and Stock Meyer. L: Some simplified NP - complete graph problems, Theoretical computer science, Vol.1563, pp. 561 - 570, 1999.

[10] Glover. F: Tabu Search - Part I, ORSA journal of computing, vol. 1, No.3, (1989), pp. 190 - 206.

[11] Glover. F: Tabu search: A Tutorial, Interface 20, pp. 74 - 94, 1990.

[12] Gomes. F. C, Meneses. C. N, Pardalos. P. M and Viana. G. V. R: Experimental analysis of approximation algorithms for the vertex cover and set covering problems, Journal of computers and Operations Research, vol. 33, pp. 3520 - 3534, 2006.

[13] Hastad. J: Some Optimal Inapproximability Results., Journal of the ACM, vol. 48, No.4, pp. 798 - 859, 2001.

[14] Hochbaum. D. S: Approximation algorithm for the set covering and vertex cover problems, SIAM Journal on computing, 11(3), 555 - 6, 1982.

[15] D. Holliday, R. Resnick, J. Walker, Fundamentals of physics, John Wiley and Sons, 1993.

[16] Johnson. D.S, Approximation Algorithms for Combinatorial problems,J.Comput.System Sci.9(1974)256-278.

[17] Johnson, D.s., C.R Aragon, L.A. McGeoch, and C. Schevon. (1989). ”Optimization by Simulated Anealing: An Experimental Evaluation, Part I: Graph Partitioning.” Operations Research 37, 875-892.

[18] Johnson, D.S., C.R. Aragon, L.A. McGeoch, and C.Schevon. (1989b). ”Optimization by Simulated Annealing: An Experimental Evaluation, part II: Graph Coloring and Number Partitioning.” Operations Research 39, 378-406.

[19] Karp. R. M: Reducibility among combinatorial problems, Plenum Press, New York, pp. 85 - 103, 1972.

[20] I.R. Kenyon, General Relativity, Oxford University Press, 1990.

[21] Khuri S, Back Th. An Evolutionary heuristic for the minimum vertex-cover problem. 18th Deutche Jahrestagung fur Kunstliche. Max-Planck Institut fur Informatik Journal 1994;MPI-I-94-241:86-90.

[22] Likas, A and Stafylopatis, A: A parallel algorithm for the minimum weighted vertex cover problem, Information Processing Letters, vol. 53, pp. 229 - 234, 1995.

[23] R. Mansouri, F. Nasseri, M. Khorrami, Effective time variation of G in a model universe with variable space dimension, Physics Letters 259 (1999) 194-200.

[24] Motwani, R. (1992). ”Lecture Notes on Application. ” Technical Report, STAN-CS-92-1435, Department of Computer Science, Stanford University.

[25] Motwani. R: Lecture Notes on Approximation Algorithms, Technical Report, STAN-CS-92-1435, Department of Computer Science, Stanford University, 1992.

[26] Neidermeier. R and Rossmanith. P: On efficient fixed-parameter algorithms for weighted vertex cover, Journal of Algorithms, vol. 47, pp. 63 - 77, 2003.

[27] Pitt. L: A Simple Probabilistic Approximation Algorithm for Vertex Cover, Technical Report, YaleU/DCS/TR-404, Department of Computer Science, Yale University, 1985.

[28] E. Rashedi, Gravitational Search Algorithm, M.Sc. Thesis, Shahid Bahonar University of Kerman, Kerman, Iran, 2007 (in Farsi).

[29] B. Schutz, Gravity from the Ground Up, Cambridge University Press, 2003.

[30] Monien. B and Speckenmeyer. E: Ramsey numbers and an approximation algorithm for the vertex cover problems, Acta Informatica, vol. 22, pp. 115 - 123, 1985.

[31] Shyu. S.J, Yin. P.Y and Lin. B.M.T: An ant colony optimization algorithm for the minimum weight vertex cover problem, Annals of Operations Research, Vol. 131, pp. 283 - 304, 2004.

[32] Weight. M and Hartmann. A. K: The number of guards needed by a museum - a phase transition in vertex covering of random graphs., Phys - Rev. Lett., 84, 6118, 2000b.

[33] Weight. M and Hartmann. A. K.: Minimal vertex covers on finite connectivity random graphs - A hard-sphere lattice-gas picture, Phys. Rev. E, 63, 056127.