Applied Mechanics and Materials Vol. 591 (2014) pp 172-175
© (2014) Trans Tech Publications, Switzerland
doi:10.4028/www.scientific.net/AMM.591.172

Submitted: 2014-05-10
Revised: 2014-06-02
Accepted: 2014-06-02
Online: 2014-07-18

Computational complexity analysis of selective breeding algorithm

M.Chandrasekaran[1,a], P.Sriramya[2,b], B.Parvathavarthini[3,c], M.Saravanamanikandan[3.d]

[1] Professor, Mechanical Engineering, Vels University, Chennai, India.

[2]Computer Science and Engineering, Saveetha School of Engineering, Chennai, India

[3]Professor, Computer Applications, St. Joseph's College of Engineering, Chennai, India

[4] Associate Professor, Mechanical Engineering, Dr.MGR University, Chennai, India.

[2]sriramya82@yahoo.com

Key words: Computational Complexity, NP-hard, Selective Breeding algorithm

Abstract . In modern years, there has been growing importance in the design, analysis and to resolve extremely complex problems. Because of the complexity of problem variants and the difficult nature of the problems they deal with, it is arguably impracticable in the majority time to build appropriate guarantees about the number of fitness evaluations needed for an algorithm to and an optimal solution. In such situations, heuristic algorithms can solve approximate solutions; however suitable time and space complication take part an important role. In present, all recognized algorithms for NP-complete problems are requiring time that's exponential within the problem size. The acknowledged NP-hardness results imply that for several combinatorial optimization problems there are no efficient algorithms that realize a best resolution, or maybe a close to best resolution, on each instance. The study Computational Complexity Analysis of Selective Breeding algorithm involves both an algorithmic issue and a theoretical challenge and the excellence of a heuristic.

Introduction

Computational complexity theory is a branch of the theory of computation in theoretical computer science and mathematics that focuses on classifying computational problems according to their inherent difficulty, and relating those classes to each other. A computational problem is understood to be a task that is in principle amenable to being solved by a computer, which is equivalent to stating that the problem may be solved by mechanical application of mathematical steps.The word "computational complexity" has two usages that should be renowned. It refers to an algorithm for resolution instances of a problem: generally explicit, the process complexness of an algorithm may be a live of what percentage steps the algorithm would force within the worst case for an instance or input of a given size. The amount of steps is measured as a perform of that size. The term's second, additional necessary use is in relevancy a drag itself. The design of procedure complexness involves classifying issues in line with their inherent flexibility or intractableness that's, whether or not they are "easy" or "hard" to unravel. This classification theme includes the well-known categories P and NP; the terms "NP-complete" and "NP-hard" are associated with the category NP.

Multi-objective optimization problems

An objective optimization drawback consists in looking an answer x € X, in order that the target functions, f(x), have a most or minimum worth. During this case, to check two people-solutions, x (1) and x (2), means that to check their objective values, f(x (1)) and f(x (2)). Multi Objective optimization problem is outlined by a group of n parameters (decision variables), a group of k objective functions and a group of m constraints. The aim of improvement is to of the choice variables [1]. The aim of optimization is to,
Maximize or Minimize $Z = f(x) = f_1(x), f_2(x),\ldots\ldots., f_k(x)$,

with satisfying the constraints $g(x) = g_1(x) , g_2(x) ,\ldots\ldots., g_m(x) \leq 0$

where x = (x_1, x_2,…, x_n) € X, y = (y_1, y_2,…, y_k) € Y.

Here x is named the decision vector, y is named the objective vector, X the decision space and Y_f = $f(X_f)$ the objective space. The feasible set X_f is the set of decision vectors x which satisfy the constraints $g(x) \leq 0$.

Computational complexity and NP-hardness

Many researchers tried cutting planes, dynamic programming, branch and bound, and group theoretical methods, but all failed to solve the medium-sized cases. Almost all real problems are either easy or NP-hard [2].There is several reasons why a tough downside would possibly still be solved in follow. NP-hard means that solely that it takes a protracted time to resolve specifically all cases of sufficiently giant size.

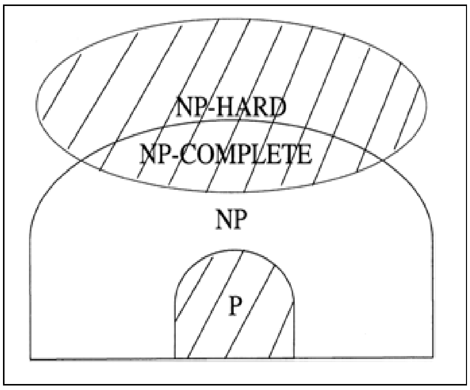

Fig 1. NP-hard Problems [2]

Selective breeding algorithm (SBA)

A new Evolutionary algorithm named Selective Breeding Algorithm (SBA) was proposed by Sriramya et al [3], where the solutions are made to breed, mutate, sort and multiple better solution are formed then fitness condition are placed and new generation of best solution are created survive, by this we can extract the best solution among them. Haploid means single chromosome/string and diploid means two chromosome/string.

STEP 1 : Initialize the population

STEP 2 : i) Find the objective function value and breeding factor for each haploid.

Breeding factor = 1/objective function value

ii) Sort the population based on breeding factor (or) objective function value

STEP 3 : i) Divide the population into two sets (i.e. first five sequences as one set called dominant set and remaining sequences as another set called recessive set).

ii) Form diploids (set of haploids) for breeding process which contains one dominant and one recessive sequence.

SET 'R' - R_1 R_2 R_3 R_4 R_5 Dominant haploid set

SET 'r' - r_1 r_2 r_3 r_4 r_5 Recessive haploid set

STEP 4: Perform breeding process for all possible combinations of diploid by taking two at a time.

Five set of diploid i.e., 1) R_1r_1 2) R_2r_2 3) R_3r_3 4) R_4r_4 5) R_5r_5

Consider two set of diploid (R_4r_4 x R_5r_5). Possible breeds are

R_4R_5	R_4r_5
R_4R_5	R_4r_5

By the same way, following diploid combinations are obtained.

STEP 5: i) Do fusion process for diploids obtained from breeding process. The possible breeds obtained while considering any two haploids is called one set. For each set, randomly select fusion points. Number of fusion points= length of the given haploid/2. At the fusion points interchange genes between parents.

ii) Divide each diploid into two haploids.

STEP 6: Selective breeding of particular genes runs the risk of losing some of the other genes from the gene pool altogether, which is irreversible. This is called in-breeding depression. To avoid this, add 10% of haploid in each iteration.

STEP 7: Sort the haploids obtained from step 2, 4 and 5 based on breeding factor/objective function value and take first 10 haploids for next iteration.

STEP 8: Goto STEP 3 and repeat the processes to the required no. of iterations.

Problem for SBA

X_{ik} is the starting time of job i on machine k, q_{ijk} is the indicator which takes on a value of 1 if operation j of job i requires machine k, and zero otherwise. Y_{ihk} is the variable which takes on a value of 1 if job i precedes job h on machine k, and zero otherwise [4]. The objective function for the given Job Shop Scheduling is

$$\text{Minimize } Z = \sum_{k=1}^{m} q_{imk} (X_{ik} + p_{ik}) \qquad (i = 1,\ldots.n)$$

Subject to

a) ***Sequence constraint***

b) ***Resource constraint***

Results and discussion

The proposed Selective Breeding algorithm (SBA) has been tested for 80 problem instances of various sizes collected in the following Eighty instances of eight different size (n X m = 15 X 15; 20 X 15; 20 X 20; 30X15; 30X 20; 50 X 15; 50 X 20; 100 X 20) denoted by Taillard (TA1-TA80) [5]. The Relative Error RE (%) was calculated for all problem instances, as a percentage by which the solution obtained is above the optimum value (Opt) if it is known or best known lower bound (LB) [6].

$$RE (\%) = 100 \text{ X } (UB – LB)/LB.$$

In Table 1, the result obtained in SBA is compared with AIS, TSSB and SB-GLS1 procedure. The mean relative error for class (C) problem instances obtained by SBA (1.628%) is lower than previously obtained results of 1.865% from AIS, 2.56% from TSSB procedure and 3.68% from SB-GLS1 procedure. But in SBA, the relative error obtained for the individual problems is low in most of the cases when compared with AIS, which reflects on the mean relative error.

Table 1. Comparison of Mean Relative Error and Computing Time of SBA with AIS, TSSB and SB-GLS1 of Balas and Vazacopoulos

Problem	n	m	SBA		AIS		TSSB		SB-GLS1	
			MRE	Tav	MRE	Tav	MRE	Tav	MRE	Tav
TA 01-10	15	15	0.763	106	0.08	118	1.45	2175	2.24	57
TA 11-20	20	15	3.004	254	3.23	232	4.13	2526	6.18	113
TA 21-30	20	20	4.829	389	5.21	495	6.52	34910	8.12	165
TA 31-40	30	15	0.966	791	1.34	835	1.92	14133	3.53	175
TA 41-50	30	20	3.675	1558	4.89	2331	6.04	11512	8.5	421
TA 51-60	50	15	0	473	0.01	665	0.02	421	0.02	152
TA 61-70	50	20	0.073	1092	0.16	1315	0.39	6342	0.83	590
TA 71-80	100	20	0	985	0.01	1019	0	231	0	851
MRE			1.628		1.87		2.56		3.68	

Conclusion

In optimization, the aim is to search out as many various solutions as potential near to optimal. As a result of there area unit many sorts of advanced complex problems and no single algorithm is that the best for each sort and even for each instance, many methods and algorithms are developed within the literature to resolve these issues. The delineated performance indicators permit to measuring the performance of an algorithm, to adjust the parameters of an algorithmic program to get higher results and additionally to check completely different algorithms. The measures are often quantitative or qualitative. The measures are often quantitative or qualitative. Computational Complexity of SBA Algorithms is NP hard based on the above case study.

References

[1] Zitzler,E., Deb, K., Lothar,T., Comparison of multiobjective evolutionary algorithms: Empirical results, Evolutionary Computation Journal, 8(2)(2000) pp. 125-148.
[2] Craig A., Tovey, Tutorial on Computational Complexity , Interfaces, Vol. 32, No. 3, May–June 2002, pp30-61
[3] P. Sriramya, B. Parvathavarthini and T. Balamurugan, “A Novel Evolutionary Selective Breeding Algorithm and its Application”, Asian Journal of Scientific Research, DOI: 10.3923/ajsr.2012.
[4] Yang S., Dingwei Wang, A new adaptive neural network and heuristics hybrid approach for job shop scheduling, Computers and operations Research, 28(2001)pp 955-971.
[5] Taillard E., Benchmarks for basic scheduling problems, European Journal of Operational Research, 64(1993) pp 278-285.
[6] Pezzella F., Emanuela Merelli, A tabu search method guided by shifting bottleneck for the job shop scheduling problem, European Journal of Operational Research 120 (2000), pp 297-310.

Advanced Manufacturing Research and Intelligent Applications
10.4028/www.scientific.net/AMM.591

Computational Complexity Analysis of Selective Breeding Algorithm
10.4028/www.scientific.net/AMM.591.172