Cultural Algorithms for the Set Covering Problem

Broderick Crawford[1,2], Ricardo Soto[1,3], and Eric Monfroy[4]

[1] Pontificia Universidad Católica de Valparaíso, Chile
FirstName.Name@ucv.cl
[2] Universidad Finis Terrae, Chile
[3] Universidad Autónoma de Chile, Chile
[4] LINA, Université de Nantes, France
FirstName.Name@univ-nantes.fr

Abstract. This paper addresses the solution of weighted set covering problems using cultural algorithms. The weighted set covering problem is a reasonably well known NP-complete optimization problem with many real world applications. We use a cultural evolutionary architecture to maintain knowledge of diversity and fitness learned over each generation during the search process. The proposed approach is validated using benchmark instances, and its results are compared with respect to other approaches which have been previously adopted to solve the problem. Our results indicate that the approach is able to produce very competitive results in compare with other algorithms solving the portfolio of test problems taken from the ORLIB.

Keywords: Weighted Set Covering Problem, Cultural Algorithm, Genetic and Evolutionary Computation.

1 Introduction

The Weighted Set Covering Problem (WSCP) is a kind of problem that can model several real life situations [7, 3]. In this work, we solve some benchmarks of WSCP with an evolutive approach: Cultural Algorithms [13–15]. Cultural Algorithms are a technique that incorporates knowledge obtained during the evolutionary process trying to make the search process more efficient. Cultural algorithms have been successfully applied to several types of optimization problems [4, 10]. However, only a few papers had proposed a cultural algorithm for SCP and solving a few instances [6].

Here, the proposed approach is validated using 45 instances and its results are compared with respect to 8 other approaches. This paper is organized as follows: In Section 2, we formally describe WSCP using mathematical programming models. In section 3 we present the Cultural Evolutionary Architecture. In sections 4 and 5 we show the Population Space and the Belief Space considered to solve WSCP with Cultural Algorithms mantaining Diversity and Fitness knowledge. In Section 6, we present experimental results obtained when applying the algorithm for solving the standard benchmarks taken from the ORLIB [1]. Finally, in Section 7 we conclude the paper.

Y. Tan, Y. Shi, and H. Mo (Eds.): ICSI 2013, Part II, LNCS 7929, pp. 27–34, 2013.

2 Problem Description

The WSCP is the NP-complete problem of partitioning a given set into subsets while minimizing a cost function defined as the sum of the costs associated to each of the eligible subsets [3]. In the WSCP matrix formulation we are given a $m \times n$ matrix $A = (a_{ij})$ in which all the matrix elements are either zero or one. Additionally, each column is given a weight (non-negative cost) c_j. We say that a column j can cover a row i if $a_{ij} = 1$. Let J denotes the set of the columns and x_j a binary variable which is one if column j is chosen and zero otherwise. The WSCP can be defined formally as follows:

$$Minimize \qquad f(x) = \sum_{j=1}^{n} c_j x_j \tag{1}$$

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1; \qquad \forall i = 1, \ldots, m \tag{2}$$

The goal in the WSCP is to choose a subset of the columns of minimal weight which covers every row.

3 Evolutionary Architecture: Cultural Algorithms

The Cultural Algorithms were developed by Robert G. Reynolds [13–15], as a complement to the metaphor used by Evolutionary Algorithms that are mainly focused on natural selection and genetic concepts. The Cultural Algorithms are based on some theories which try to model cultural as an inheritance process operating at two levels: a *Micro-evolutionary level*, which consists of the genetic material that an offspring inherits from its parent, and a *Macro-evolutionary level*, which is the knowledge acquired by the individuals through generations. This knowledge, once encoded and stored, it serves to guide the behavior of the individuals that belong to a population. Considering that evolution can be seen like an optimization process, Reynolds developed a computational model of cultural evolution that can have applications in optimization [4, 10]. He considered the phenomenon of double inheritance with the purpose of increase the learning or convergence rates of an evolutionary algorithm. In this model each one of the levels is represented by a space. The micro-evolutionary level is represented by the Population Space and the macro-evolutionary level by the Belief Space.

The *Population Space* can be adopted by anyone of the paradigms of evolutionary computation, in all of them there is a set of individuals where each one has a set of independent characteristics with which it is possible to determine his aptitude or fitness. Through time, such individuals could be replaced by some of their descendants, obtained from a set of operators (crossover and mutation, for example) applied to the population.

The *Belief Space* is the ”store of the knowledge” acquired by the individuals along the generations. The information in this space must be available for the population of individuals. There is a protocol of communication established to dictate rules about the type of information that it is necessary to interchange between the spaces. This protocol defines two functions: *Acceptance*, this function extracts the information (or experience) from the individuals of a generation putting it into the Belief Space; and *Influence*, this function is in charge "to influence" in the selection and the variation operators of the individuals (as the crossover and mutation in the case of the genetic algorithms). This means that this function exerts a type of pressure according to the information stored in the Belief Space.

3.1 Types of Knowledge

Knowledge that are important in the Belief Space of any cultural evolution model: Situational, Normative, Topographic, Historical or Temporal, and Domain Knowledge. According to Reynolds and Bing [15, 12], they conform a complete set, that is any other type of knowledge that is desired to add can be generated by means of a combination of two or more of the previous types of knowledge. The pseudo-code of a cultural algorithm is shown in Algorithm 1 [9]. Most of the steps of a cultural algorithm correspond with the steps of a traditional evolutionary algorithm. It can be clearly seen that the main difference lies in the fact that cultural algorithm use a Belief Space. In the main loop of the algorithm, we have the update of the belief space. It is at this point in which the belief space incorporates the individual experiences of a select group of members with the acceptance function, which is applied to the entire population.

Algorithm 1. Sketch of the Cultural Algorithm

Generate the initial population (*Population Space*)
Initialize the Belief Space
Evaluate the initial population
repeat
    Update the Belief Space (*Acceptance function*)
    Apply the variation operators (*considering Influence function*)
    Evaluate each child
    Perform selection
until *the end condition is satisfied*

4 Micro-evolutionary Level: Population Space

In the design and development of our cultural algorithm solving WSCP we considered in the Population Space a Genetic Algorithm with binary representation. An individual, solution or chromosome is a n-bit string, where a value 1 in the bit indicates that the column is in the solution and zero in another case. The initial population was generated with *size of the population* selected individuals randomly with a repair process in order to assure the feasibility of the individuals. For the selection of parents we used tournament. For the process of variation we used the operator of fusion crossover proposed by Beasley and

Chu [2], for mutation we used interchange and multibit. For the treatment of not feasible individuals we applied the repairing heuristic proposed by Beasley and Chu too [2]. In the replacement of individuals we use the strategy steady state and the heuristic proposed by Lozano et al. [11], which is based on the level of diversity contribution of the new offspring. The genetic diversity was calculated by the Hamming distance, which is defined as the number of bit differences between two solutions. The main idea is try to replace a solution with worse fitness and with lower contribution of diversity than the one provided by the offspring. In this way, we are working with two underlying objectives simultaneously: to optimize the fitness and to promote useful diversity.

5 Macro-evolutionary Level: Belief Space

In the cultural algorithm, the shared belief space is the foundation on which the efficiency of the search process depends. In order to find better solutions and improve the convergence speed we incorporated information about the diversity in the Belief Space. We stored in the Belief Space the individual with better fitness of the current generation and the individual who delivers major diversity to the population. With this type of knowledge Situational and Historical or Temporal, each one of the new individuals generated tries to follow a leader.

Initializing the Belief Space. A Situational-Fitness knowledge procedure selects from the initial population the individual with better fitness, which will be a leader in the Situational-Fitness space of beliefs. A Situational-Diverse knowledge procedure selects from the initial population the most diverse individual of the population, which will be a leader in the Situational-Diverse space of beliefs.

Applying the Variation Operators. Here we implemented the Influence of Situational-Fitness knowledge in the operator of Crossover. The influence initially appears at the moment of the election of the parents, the father 1 will be chosen with the method of binary tournament and the father 2 will be the individual with better fitness stored in the space of beliefs. Influence of Situational-Diverse knowledge in the operator of Crossover. This procedure works recombining the individual with better fitness of every generation with the most diverse stored in the space of beliefs, with this option we expect to deliver diversity to the population.

Updating the Belief Space. Updating the Situational Belief Space procedure, it implies that the Situational space of beliefs will be updated in all generations of the evolutionary process. The update of the Situational space of beliefs consists in the replacement of the individuals by current generation individuals if they are better considering Fitness and Diversity.

6 Experiments and Results

The performance of the algorithm was evaluated experimentally solving WSCP benchmarks from ORLIB [1]. Table 1 shows their detailed information. The first

Table 1. Problem instances

Problem	Number of constraints (m)	Number of variables (n)	Density(%)	Cost range	Best-known solution
wscp41	200	1000	2	[1-100]	429
wscp42	200	1000	2	[1-100]	512
wscp43	200	1000	2	[1-100]	516
wscp44	200	1000	2	[1-100]	494
wscp45	200	1000	2	[1-100]	512
wscp46	200	1000	2	[1-100]	560
wscp47	200	1000	2	[1-100]	430
wscp48	200	1000	2	[1-100]	492
wscp49	200	1000	2	[1-100]	641
wscp410	200	1000	2	[1-100]	514
wscp51	200	2000	2	[1-100]	253
wscp52	200	2000	2	[1-100]	302
wscp53	200	2000	2	[1-100]	226
wscp54	200	2000	2	[1-100]	242
wscp55	200	2000	2	[1-100]	211
wscp56	200	2000	2	[1-100]	213
wscp57	200	2000	2	[1-100]	293
wscp58	200	2000	2	[1-100]	288
wscp59	200	2000	2	[1-100]	279
wscp510	200	2000	2	[1-100]	265
wscp61	200	1000	5	[1-100]	138
wscp62	200	1000	5	[1-100]	146
wscp63	200	1000	5	[1-100]	145
wscp64	200	1000	5	[1-100]	131
wscp65	200	1000	5	[1-100]	161
wscpa1	300	3000	2	[1-100]	253
wscpa2	300	3000	2	[1-100]	252
wscpa3	300	3000	2	[1-100]	232
wscpa4	300	3000	2	[1-100]	234
wscpa5	300	3000	2	[1-100]	236
wscpb1	300	3000	5	[1-100]	69
wscpb2	300	3000	5	[1-100]	76
wscpb3	300	3000	5	[1-100]	80
wscpb4	300	3000	5	[1-100]	79
wscpb5	300	3000	5	[1-100]	72
wscpc1	400	4000	2	[1-100]	227
wscpc2	400	4000	2	[1-100]	219
wscpc3	400	4000	2	[1-100]	243
wscpc4	400	4000	2	[1-100]	219
wscpc5	400	4000	2	[1-100]	215
wscpd1	400	4000	5	[1-100]	60
wscpd2	400	4000	5	[1-100]	66
wscpd3	400	4000	5	[1-100]	72
wscpd4	400	4000	5	[1-100]	62
wscpd5	400	4000	5	[1-100]	61

column presents the problem code, the second and third columns show the number of constraints (m) and the number of variables (n). The fourth column shows the density (it is the percentage of non-zero entries in the WSCP matrix). The fifth column shows the range of costs of the variables. The last column presents the best known solution for each instance. Table 2 presents the results (the best cost obtained) when applying our algorithm for solving the WSCP benchmarks. The first two columns present the problem code and the best known solution for each instance. The following columns show the results applying Ant System (AS) and Ant Colony System (ACS) taken from [5] and Round, Dual-LP, Primal-Dual, Greedy taken from [8]. The next columns show the costs from Genetic Algorithms *(GA-1 and GA-2 using only the micro-evolutionary level)* with the basic proposal described in 4 without considering diversity. And the costs obtained by Cultural Algorithms *(CA-1 and CA-2 using micro and macro-evolution).* With GA-1 and

Table 2. Cost obtained using different algorithms

Problem	Best-known	AS	ACS	Round	Dual-LP	Primal-Dual	Greedy	GA-1	CA-1	GA-2	CA-2
wscp41	429	473	463	429	505	521	463	506	462	448	448
wscp42	512	594	590					609	582	642	603
wscp43	516							554	591	532	540
wscp44	494							551	577	512	512
wscp45	512							531	545	527	520
wscp46	560							574	620	568	605
wscp47	430							458	483	437	447
wscp48	492	524	522		522		499	560	549	609	548
wscp49	641							700	763	675	671
wscp410	514			539	664	669	556	548	596	553	533
wscp51	253	289	280	405	324	334	293	298	296	380	309
wscp52	302							353	335	320	330
wscp53	226							264	245	239	232
wscp54	242							281	265	244	250
wscp55	211							245	230	219	218
wscp56	213							243	224	232	227
wscp57	293							328	314	309	310
wscp58	288							326	315	306	311
wscp59	279							325	285	292	292
wscp510	265							292	280	277	278
wscp61	138	157	154	301	210	204	155	172	156	162	155
wscp62	146	169	163	347	209	232	170	162	162	188	171
wscp63	145	161	157				167	170	164	178	176
wscp64	131							145	138	133	141
wscp65	161							183	181	179	186
wscpa1	253			592	331	348	288	319	263	253	303
wscpa2	252			531	376	378	285	289	266	267	272
wscpa3	232			473	295	319	270	267	261	245	245
wscpa4	234			375	301	333	278	257	257	247	251
wscpa5	236			349	335	353	272	262	247	239	248
wscpb1	69			196	115	101	75	102	95	101	87
wscpb2	76			243	110	117	87	118	84	81	78
wscpb3	80			207	117	112	89	119	87	83	85
wscpb4	79							123	89	84	83
wscpb5	72							106	79	77	75
wscpc1	227			442	317	305	261	260	260	308	254
wscpc2	219			484	311	309	260	281	241	233	225
wscpc3	243			551	328	367	268	302	270	264	259
wscpc4	219			523	303	324	259	265	240	237	240
wscpc5	215							271	233	219	219
wscpd1	60			184	105	92	72	139	69	62	68
wscpd2	66			209	113	96	74	157	69	70	71
wscpd3	72			221	119	111	83	149	78	79	80
wscpd4	62							151	68	66	67
wscpd5	61							151	65	64	66

CA-1 we used Fusion Crossover and Mutation Interchange. With GA-2 and CA-2 we used Fusion Crossover and Mutation MultiBit. The algorithm has been run with the following parameters setting: size of the population (n)=100, size of the tournament (t)=2, number of generations (g)=30, with interchange and multibit mutation we affect the 5‰of the bits with a probability of mutation (p_m)=0.2. The algorithm were implemented using ANSI C, GCC 3.3.6, under Microsoft Windows XP Professional version 2002.

It is possible to observe that the incorporation of diversity in the Genetic Algorithm produced an improvement in the performance of the algorithm. The results indicate that the approach is able to produce very competitive results in compare with other approximation algorithms solving the portfolio of test problems taken from the ORLIB.

7 Conclusions

In this paper, we propose the use of knowledge of diversity to improve the performance of an evolutionary algorithm when solving the set covering problem. The executed experiments provided encouraging results in compare with other approaches. Our computational results confirm that incorporating information about the diversity of solutions we can obtain good results in the majority of the experiments. Our main conclusion from this work is that we can improve the performance of genetic algorithms considering additional information in the evolutionary process.

By other side, genetic algorithms tends to lose diversity very quickly. In order to deal with this problem, we have shown that maintaining diversity in the belief space we can improve the computational efficiency.

References

1. Beasley, J.E.: Or-library:distributing test problem by electronic mail. Journal of Operational Research Society 41(11), 1069–1072 (1990), `http://people.brunel.ac.uk/~mastjjb/jeb/info.html`
2. Beasley, J.E., Chu, P.C.: A genetic algorithm for the set covering problem. European Journal of Operational Research 94(2), 392–404 (1996)
3. Chu, P.C., Beasley, J.E.: Constraint handling in genetic algorithms: The set partitioning problem. Journal of Heuristics 4(4), 323–357 (1998)
4. Coello, C.A., Landa, R.: Constrained Optimization Using an Evolutionary Programming-Based Cultural Algorithm. In: Parmee, I. (ed.) Proceedings of the Fifth International Conference on Adaptive Computing Design and Manufacture (ACDM 2002), University of Exeter, Devon, UK, vol. 5, pp. 317–328. Springer (April 2002)
5. Crawford, B., Castro, C.: Integrating lookahead and post processing procedures with aco for solving set partitioning and covering problems. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 1082–1090. Springer, Heidelberg (2006)
6. Crawford, B., Lagos, C., Castro, C., Paredes, F.: A cultural algorithm for solving the set covering problem. In: Melin, P., Castillo, O., Gómez-Ramírez, E., Kacprzyk, J., Pedrycz, W. (eds.) Analysis and Design of Intelligent Systems using Soft Computing Techniques. ASC, vol. 41, pp. 408–415. Springer, Heidelberg (2007)
7. Feo, T., Resende, M.: A probabilistic heuristic for a computationally difficult set covering problem. Operations Research Letters 8, 67–71 (1989)
8. Gomes, F.C., Meneses, C.N., Pardalos, P.M., Viana, G.V.R.: Experimental analysis of approximation algorithms for the vertex cover and set covering problems. Comput. Oper. Res. 33(12), 3520–3534 (2006)
9. Landa, R., Coello, C.A.: Optimization with constraints using a cultured differential evolution approach. In: GECCO 2005: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, pp. 27–34. ACM Press, New York (2005)
10. Landa, R., Coello, C.A.: Use of domain information to improve the performance of an evolutionary algorithm. In: GECCO 2005: Proceedings of the 2005 Workshops on Genetic and Evolutionary Computation, pp. 362–365. ACM Press, New York (2005)

11. Lozano, M., Herrera, F., Cano, J.R.: Replacement strategies to preserve useful diversity in steady-state genetic algorithms. In: Proceedings of the 8th Online World Conference on Soft Computing in Industrial Applications (September 2003)
12. Peng, B.: Knowledge and population swarms in cultural algorithms for dynamic environments. PhD thesis, Detroit, MI, USA, Adviser-Reynolds, R.G. (2005)
13. Reynolds, R.: An introduction to cultural algorithms. In: Third Annual Conference on Evolutionary Programming, pp. 131–139 (1994)
14. Reynolds, R.G.: Cultural algorithms: theory and applications. In: New Ideas in Optimization, pp. 367–378. McGraw-Hill Ltd., Maidenhead (1999)
15. Reynolds, R.G., Peng, B.: Cultural algorithms: Modeling of how cultures learn to solve problems. In: ICTAI, pp. 166–172. IEEE Computer Society (2004)