Using the Firefly Optimization Method to Solve the Weighted Set Covering Problem

Broderick Crawford[1,2], Ricardo Soto[1,3], Miguel Olivares-Suárez[1], and Fernando Paredes[4]

[1] Pontificia Universidad Católica de Valparaíso, Chile
[2] Universidad Finis Terrae, Chile
[3] Universidad Autónoma de Chile, Chile
[4] Escuela de Ingeniería Industrial, Universidad Diego Portales, Chile
{broderick.crawford,ricardo.soto}@ucv.cl,
miguel.olivares.s@mail.pucv.cl,
fernando.paredes@udp.cl

Abstract. The Weighted Set Covering Problem is a formal model for many practical optimization problems. In this problem the goal is to choose a subset of columns of minimal cost covering every row. Here, a Binary Firefly Algorithm has been developed to tackle the Weighted Set Covering Problem. Firefly Algorithm is a recently developed population-based metaheuristic inspired by the flashing behaviour of fireflies. Experimental results show that our approach is competitive solving the problem at hand.

Keywords: Weighted Set Covering Problem, Firefly Algorithm, Swarm Intelligence.

1 Introduction

The Weighted Set Covering Problem (WSCP) has many applications, including those involving routing, scheduling, stock cutting, electoral redistricting and others important real life situations [14]. Different solving methods have been proposed in the literature for the Weighted Set Covering Problem. Exact algorithms are mostly based on Branch-and-Bound and Branch-and-Cut techniques [2,15,5], Linear Programing and Heuristic methods [7]. However, these algorithms are rather time consuming and can only solve instances of very limited size. For this reason, many research efforts have been focused on the development of metaheuristics to find as result good or near-optimal solutions within a reasonable period of time. An incomplete list of metaheuristics for the WSCP includes Genetic Algorithms [1,3], Simulated Annealing [6], Tabu Search [8], Cultural Algorithms [12,10] and Ant Colony Optimization [9]. For a deeper comprehension of most of the effective algorithms for the WSCP in the literature, we refer the interested reader to the survey by [7]. In this paper we present the metaheuristic Firefly Algorithm that is relatively new in the area to solve the WSCP.

C. Stephanidis (Ed.): HCII 2014 Posters, Part I, CCIS 434, pp. 509–514, 2014.

2 Firefly Algorithm

The Firefly Algorithm (FA) is an algorithm inspired by the social behavior of fireflies. For more details of FA we refer the reader to [16,17]. The pseudo code of the firefly-inspired algorithm was developed using the following rules: All fireflies are unisex and are attracted to other fireflies regardless of their sex. The degree of the attractiveness of a firefly is proportional to its brightness, and thus for any two flashing fireflies, the one that is less bright will move towards to the brighter one. Finally, the brightness of a firefly is determined by the value of the objective function. For a maximization problem, the brightness of each firefly is proportional to the value of the objective function. In the original FA, the result of applying a movement (generating the new dimension value of the firefly) is probable to be a real number, to fix this for WSCP we occupy the following binarization rule:

$$x_i^k(t+1) := \begin{cases} 1 \text{ if } rand < T(x_i^k(t+1)) \\ 0 \;\; \text{otherwise} \end{cases} \tag{1}$$

where $rand$ is a uniform random number between 0 and 1, $x_i^k(t+1)$ is the value resulting from the traditional FA movement and $T(x)$ is the binary transfer function. A transfer function defines the probability of changing a position vector elements from 0 to 1 and vice versa. Transfer functions force firefly bits to move in a binary space. In this work the transfer function used is $T(x) = |\frac{2}{\pi}\arctan(\frac{\pi}{2}x)|$. Algorithm 1 shows the pseudo code of FA.

Algorithm 1: Pseudo code of Binary FA for WSCP

```
Begin
  Initialize parameters
  Evaluate the light intensity I determined by f(x) Eq. 2
  while t < MaxGeneration do
    for i = 1 : m (m fireflies) do
      for j = 1 : m (m fireflies) do
        if (I_j < I_i) then
          movement = calculates value according to FA movement Eq.
          if (rand() < arctan(movement)) then
            fireflies[i][j] = 1
          else
            fireflies[i][j] = 0
          end if
        end if
        Repair solutions
        Update attractiveness
        Update light intensity
      end for j
    end for i
    t = t + 1
  end while
  Output the results
End
```

3 Weighted Set Covering Problem

In the WSCP matrix formulation we define 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 non-negative cost c_j. We say that a column j covers a row i if $a_{ij} = 1$. Let x_j a binary variable which is one if column j is chosen and zero otherwise. The WSCP can be defined formally as *minimize* (2) *subject to* (3). The goal in the WSCP is to choose a subset of the columns of minimal weight formally using constraints to enforce that each row is covered by at least one column.

$$f(x) = \sum_{j=1}^{n} c_j x_j \tag{2}$$

$$\sum_{j=1}^{n} a_{ij} x_j \geq 1; \qquad \forall i = 1, ..., m \tag{3}$$

$$x_j \in \{0, 1\}; \qquad \forall j = 1, ..., n \tag{4}$$

4 Experimental Results

The performance of Binary FA was evaluated experimentally using WSCP test instances from OR-Library of Beasley [4]. The algorithm was coded in C in NetBeans IDE 7.3 with support for C/C++ and run on a PC with a 1.8 GHz Intel Core 2 Duo T5670 CPU and 3.0 GB RAM, under Windows 8 System. In all experiments, the binary FA is executed 50 generations, and 30 times each instance. We used a population of 25 fireflies. The parameters γ, β_0 are initialized to 1. These parameters were selected empirically after a large number of tests showing good results but may not be optimal for all instances. Table 1 shows the results obtained with the 65 instances. Column "Opt." reports the optimal or the best known solution value of each instance. Columns "Min.", "Max." and "Avg." reports the minimum, maximum, and average of the best solutions obtained in the 30 executions.

Table 1. Computational results on 65 instances of WSCP

Instance	Opt.	Min.	Max.	Avg.	Instance	Opt.	Min.	Max.	Avg.
4.1	429	481	482	481.03	B.4	79	98	99	98.03
4.2	512	580	580	580.00	B.5	72	87	87	87.00
4.3	516	619	620	619.03	C.1	227	279	279	279.00
4.4	494	537	537	537.00	C.2	219	272	272	272.00
4.5	512	609	609	609.00	C.3	243	288	288	288.00
4.6	560	653	653	653.00	C.4	219	262	262	262.00
4.7	430	491	492	491.07	C.5	215	262	263	262.07
4.8	492	565	565	565.00	D.1	60	71	71	71.00
4.9	641	749	750	749.03	D.2	66	75	75	75.00
4.10	514	550	550	550.00	D.3	72	88	88	88.00
5.1	253	296	297	296.03	D.4	62	71	71	71.00
5.2	302	372	372	372.00	D.5	61	71	71	71.00
5.3	226	250	250	250.00	NRE.1	29	32	33	32.03
5.4	242	277	278	277.07	NRE.2	30	36	36	36.00
5.5	211	253	253	253.00	NRE.3	27	35	35	35.00
5.6	213	264	265	264.03	NRE.4	28	34	34	34.00
5.7	293	337	337	337.00	NRE.5	28	34	34	34.00
5.8	288	326	326	326.00	NRF.1	14	17	18	17.03
5.9	279	350	350	350.00	NRF.2	15	17	17	17.00
5.10	265	321	321	321.00	NRF.3	14	21	21	21.00
6.1	138	173	174	173.03	NRF.4	14	19	19	19.00
6.2	146	180	181	180.07	NRF.5	13	16	16	16.00
6.3	145	160	160	160.00	NRG.1	176	230	231	230.03
6.4	131	161	161	161.00	NRG.2	154	191	191	191.00
6.5	161	186	186	186.00	NRG.3	166	198	198	198.00
A.1	253	285	285	285.00	NRG.4	168	214	214	214.00
A.2	252	285	286	285.07	NRG.5	168	223	223	223.00
A.3	232	272	272	272.00	NRH.1	63	85	86	85.07
A.4	234	297	297	297.00	NRH.2	63	81	82	81.03
A.5	236	262	262	262.00	NRH.3	59	76	76	76.00
B.1	69	80	81	80.03	NRH.4	58	75	75	75.00
B.2	76	92	92	92.00	NRH.5	55	68	68	68.00
B.3	80	93	93	93.00					

5 Conclusion

We have presented a Binary Firefly Algorithm for the Weighted Set Covering Problem. As can be seen from the results obtained the metaheuristic behaves in a good way in almost all instances. This paper has demonstrated that Binary FA is an alternative to solve the WSCP, although its scope of action is continuous optimization. An interesting research direction to pursue in future work about the integration of Autonomous Search in the solving process, which in many cases has demonstrated excellent results [11,13].

Acknowledgements. Broderick Crawford is supported by Grant CONICYT/FONDECYT/1140897. Ricardo Soto is supported by Grant CONICYT/FONDECYT/INICIACION/11130459. Fernando Paredes is supported by Grant CONICYT/FONDECYT/1130455.

References

1. Aickelin, U.: An indirect genetic algorithm for set covering problems. Journal of the Operational Research Society 53, 1118–1126 (2002)
2. Balas, E., Carrera, M.C.: A dynamic subgradient-based branch-and-bound procedure for set covering. Operations Research 44(6), 875–890 (1996)
3. Beasley, J., Chu, P.: A genetic algorithm for the set covering problem. European Journal of Operational Research 94(2), 392–404 (1996)
4. Beasley, J.E.: A lagrangian heuristic for set-covering problems. Naval Research Logistics (NRL) 37(1), 151–164 (1990)
5. Beasley, J.E., Jornsten, K.: Enhancing an algorithm for set covering problems. European Journal of Operational Research 58(2), 293–300 (1992)
6. Brusco, M.J., Jacobs, L.W., Thompson, G.M.: A morphing procedure to supplement a simulated annealing heuristic for cost- and coverage-correlated set-covering problems. Annals of Operations Research 86, 611–627 (1999)
7. Caprara, A., Toth, P., Fischetti, M.: Algorithms for the set covering problem. Annals of Operations Research 98, 353–371 (2000)
8. Caserta, M.: abu search-based metaheuristic algorithm for large-scale set covering problems. In: Doerner, K.F., et al. (eds.) Metaheuristics: Progress in complex systems optimization. Springer
9. Crawford, B., Castro, C., Monfroy, E.: A New ACO Transition Rule for Set Partitioning and Covering Problems. In: SoCPaR, pp. 426–429 (2009)
10. Crawford, B., Lagos, C., Castro, C., Parede, F.: A Evolutionary Approach to Solve Set Covering. In: ICEIS, vol. 2, pp. 356–363 (2007)
11. Crawford, B., Soto, R., Castro, C., Monfroy, E.: Extensible CP-based autonomous search. In: Stephanidis, C. (ed.) Posters, Part I, HCII 2011. CCIS, vol. 173, pp. 561–565. Springer, Heidelberg (2011)
12. Crawford, B., Soto, R., Monfroy, E.: Cultural algorithms for the set covering problem. In: Tan, Y., Shi, Y., Mo, H. (eds.) ICSI 2013, Part II. LNCS, vol. 7929, pp. 27–34. Springer, Heidelberg (2013)
13. Crawford, B., Soto, R., Monfroy, E., Palma, W., Castro, C., Paredes, F.: Parameter tuning of a choice-function based hyperheuristic using particle swarm optimization. Expert Systems with Applications 40(5), 1690–1695 (2013)

14. Feo, T.A., Resende, M.G.C.: A probabilistic heuristic for a computationally difficult set covering problem. Operations Research Letters 8(2), 67–71 (1989)
15. Fisher, M.L., Kedia, P.: Optimal solution of set covering/partitioning problems using dual heuristics. Management Science 36(6), 674–688 (1990)
16. Yang, X.-S.: Nature-Inspired Metaheuristic Algorithms. Luniver Press (2008)
17. Yang, X.-S.: Firefly algorithms for multimodal optimization. In: Watanabe, O., Zeugmann, T. (eds.) SAGA 2009. LNCS, vol. 5792, pp. 169–178. Springer, Heidelberg (2009)