Collaborative Opposite Strategies for HS: The case of MKP

Nicolás Rojas-Morales, María-Cristina Riff Rojas, Elizabeth Montero Ureta
Universidad Técnica Federico Santa María
Valparaíso, Chile
nicolasrojas@acm.org, maria.cristina.riff@gmail.com, elizabeth.montero@usm.cl

***Abstract*—We propose a collaborative framework that considers three opposite strategies using an ants algorithm and give this information to a Harmony Search algorithm. This opposite information is used to set-up the initial Harmony Memory. We evaluate this strategy into the Adaptive Binary Harmony Search (*ABHS*), an Harmony Search approach for solving the well-known Multidimensional Knapsack Problem. Results in benchmark two instance sets show that the opposite information allows HS to focus its search to more promising areas of the search space obtaining quality solutions.**

I. Introduction

Three different opposite learning strategies have been recently proposed [1], [2], with the objective of an early detection of candidate solutions that could not be interesting to be explored by a metaheuristic, in terms of evaluation function. In this work we propose to use an ants algorithm that learns from the opposite, in order to be able to detect this kind of solutions. The information obtained is given to Adaptive Binary Harmony Search (*ABHS*) [3], an Harmony Search (HS) approach for solving Multidimensional Knapsack Problem (MKP).

The MKP is defined as a knapsack with multiple resource constraints and a set of objects $O = \{o_1, \ldots, o_N\}$, each one with a defined profit and weight in each dimension [4]. The problem consists in selecting a subset of $n \leq N$ objects in such a way that each capacity constraint, for the M dimensions, is satisfied maximizing the total profit.

The main inspiration of our work is the field of Opposite Learning (OL). In general, opposite information is applied with the objective of mapping candidate solutions for increasing the coverage of the solution space, accuracy and convergence of the search process [5]. OL has been extensively applied in Metaheuristics like Harmony Search, Particle Swarm Optimization, Differential Evolution, among others [6], [7]. In the following section, we briefly detailed the baseline algorithm named ABHS.

II. Adaptive Binary Harmony Search

Adaptive Binary Harmony Search (*ABHS*) [3] is an HS approach proposed for solving problems using a binary representation. The initial population is randomly generated. In each iteration, ABHS construct a solution X^{NEW}. For each bit, its value can be a copy from the Harmony Memory or, a randomly selected value. This decision is controlled by the *HMCR* parameter. Then, X^{NEW} can be modified using the Pitch Adjusting component. Here, with a probability of *PAR*, the algorithm can copy a bit from the best solution found to the new candidate solution. Finally, X^{NEW} can replace the worst solution in memory if it has a better quality value and, the best solution in the Harmony Memory is updated. In the following section, we present our approach inspired by the opposite learning literature.

TABLE I
Features of Opposite Learning Strategies

Strategy	Antipheromone	Heuristic Knowledge
SOL	Lowest quality solution at each iteration	Same as original algorithm
HOL	Defined for the specific problem	Modified for the specific problem
WOL	Lowest quality solution at each iteration	Inverts the original heuristic knowledge

III. Including Opposite Learning in ABHS

During the search process of a metaheuristic, there are some solutions that appear promising in the construction process, but that finally do not yield good quality solutions. We define these kind of solutions as *Quasi-near-optimal (Q-n-o)* solutions and, our work is focused in its early detection. In [1], [2], three Opposite Learning (OL) strategies were proposed to detect these *Q-n-o* solutions using ants algorithms. Here, the search process is divided into two steps: A first step where opposite learning is used to detect candidate solutions that appear interesting, but not yield to high quality solutions and, a second step where a metaheuristic algorithm incorporates this knowledge to solve the original problem. As in ACO algorithms, the learning about promising quality solutions is transmitted using a pheromone matrix. To emphasize the idea of opposite information, the pheromone used in the first step is called *antipheromone* and the first step is named *Opposite ACO*. Table I shows the details of these three OL strategies: *Soft Opposite Learning (SOL)*, *Half Opposite Learning (HOL)* and *Worst Opposite Learning (WOL)*.

In this work, the first step is performed by an ants algorithm and the second step by ABHS. Figure 1 shows the details of this scheme. The first step will search for *Q-n-o* solutions using these three strategies. Each strategy will generate an

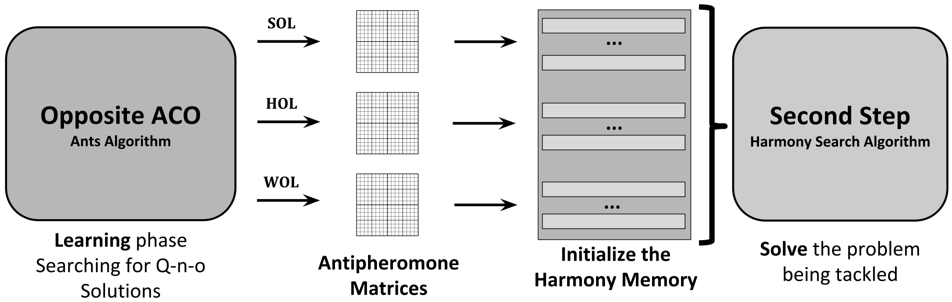

Fig. 1. Division of the search process

antipheromone matrix that will be transferred to ABHS. The information of each cell represents the desirability of choosing an object o_{i+1} when the last selected object is o_i. The higher the value of antipheromone between object o_i and o_{i+1}, the higher the number of *Q-n-o* candidate solutions that incorporated both objects. This opposite information will be used during the initialization of the Harmony Memory. We used each matrix for initialize a third of the Harmony Memory. For the construction of initial feasible candidate solutions, a greedy procedure will include objects considering different myopic functions. The possible myopic functions are:

- $Efficiency/Antipheromone$ function searches a trade-off between efficiency and antipheromone. Using this function the initialization procedure between two objects with same efficiency will prefer to include the objective with a lower amount of antipheromone.
- $Efficiency + 1/Antipheromone$ function gives more importance to efficiency than the first function. However, it takes into account antipheromone amount to decide. Here, objects lowest with efficiency have less opportunities to be chosen.
- $Efficiency$ function considers only the profit and weight in each dimension, of each object.

The key idea of these myopic functions is to construct a diverse Harmony Memory considering information about the features of each object related to their own profit and weight and, also, how related it was with *Q-n-o* candidate solutions during the *Opposite ACO*.

Finally, total resources are divided in the two steps. A budget B defines the portion of effort that will be invested in the *Opposite ACO*. Considering $maxRes$, a maximum number of evaluation functions, a portion of these will be consumed by the *Opposite ACO*. This portion is defined by the parameter $B \in [0, 1]$. Hence, the higher the budget for the *Opposite ACO*, the lower the effort during the HS step. In the following section, we present the evaluation of these strategies for solving the Multidimensional Knapsack Problem (MKP).

IV. Experiments and Results

To analyze the real contribution of the opposite information, we considered other approach named *GR-ABHS* that generates the initial Harmony Memory only focused on the Efficiency of objects. This section presents experiments to compare three approaches: *ABHS*, *GR-ABHS* and *OL-ABHS*. The objective of these tests is to evaluate the effect of including opposite information in an Harmony Search approach. For this, we considered two sets of benchmarks instances proposed by Chu and Beasley[1]: 30 instances of the set named *5x100* (5 dimensions and 100 objects) and 30 instances of the set named *10x100* (10 dimensions and 100 objects).

About the parameter values used for these experiments, we considered the same values presented by the authors. *OL-ABHS* uses the well-known Ant Knapsack (AK) [9] algorithm to search for *Q-n-o* candidate solutions. Also, the parameters values of AK are N_{Total} = 30, τ_{max} = 4 and τ_{min} = 0.01. The parameter values for *ABHS* are: HMS = 47, $HMCR$ = 0.8, PAR = 0.547 and RAN = 0.5 [10]. About the Budget B parameter, we evaluate different values ranging from 0.01 to 0.09. Finally, we obtain better results with $B = 0.02$ for the three strategies.

For these experiments we considered 50 different seeds, each one with 200.000 evaluations. As $B = 0.02$, this means that $200.000 * 0.02 = 4000$ evaluations will be used for the *Opposite ACO*. Table II shows the results for the set $5x100$ and, table III for the set $10x100$. Light grey cells represent when an algorithm outperforms other approaches considering the average quality (AVG) of the 50 seeds. On the other hand, dark grey cells represent when an algorithm outperforms other approaches considering the Best quality solution obtained.

First, results for both sets show that the search process of *ABHS* has been improved using the opposite information and also, using only the Efficency. Only in two instances (number 9 and 26 for set $10x100$), *ABHS* reached a best quality solution. About the performance in set *5x100*, *OL-ABHS* obtained better average results in 17 of the 30 instances and, *GR-ABHS* in 13 of 30. About the best quality solutions founded, *OL-ABHS* reached best solutions in 22 of the 30 instances and, *GR-ABHS*

[1] Available in http://bit.ly/instancesMKP [8]

TABLE II
RESULTS FOR SET 5X100

#	ABHS		GR-ABHS		OL-ABHS	
	AVG	BEST	AVG	BEST	AVG	BEST
1	23574,02	23780	**23833,62**	24206	23775,12	24168
2	23349,08	23652	23995,18	24225	**24031,40**	24225
3	22686,42	23051	23263,68	23470	**23270,34**	23518
4	22905,78	23057	**22979,78**	23194	22950,94	23164
5	23261,78	23560	23438,76	23636	**23454,62**	23663
6	23977,54	24319	24482,36	24555	**24504,90**	24575
7	24615,70	24865	24967,74	25244	**24990,18**	25261
8	22629,98	22912	**23095,62**	23332	23078,76	23270
9	23245,00	23604	23521,66	23846	**23536,62**	23911
10	23830,74	24190	24103,04	24259	**24105,96**	24259
11	42092,44	42409	42529,24	42705	**42567,86**	42705
12	41848,16	42119	42179,74	42279	**42222,76**	42445
13	41421,92	41707	**41561,58**	41783	41559,32	41794
14	44509,86	44904	44581,88	44818	**44606,30**	44933
15	41604,18	41915	**41997,52**	42059	41990,26	42123
16	42315,98	42644	42537,40	42657	**42578,66**	42781
17	41434,72	41676	41528,06	41746	**41602,14**	41784
18	44352,60	44569	44526,52	44780	**44586,84**	44855
19	42574,36	42965	42897,48	43279	**42904,30**	43171
20	44030,84	44337	44358,56	44511	**44373,84**	44511
21	59381,58	59692	**59599,00**	59798	59593,40	59822
22	61615,94	61825	**61753,76**	61932	61722,82	61970
23	59427,82	59689	**59594,28**	59726	59548,14	59704
24	60078,10	60366	60274,56	60417	**60275,70**	60409
25	60724,62	61029	**60996,66**	61055	60903,16	61079
26	58504,70	58689	**58742,12**	58920	58669,68	58937
27	61135,08	61365	**61315,90**	61538	61270,08	61522
28	61123,72	61403	**61202,16**	61425	61173,80	61437
29	58987,64	59219	59100,18	59314	**59137,52**	59380
30	59621,38	59908	**59849,06**	59933	59805,68	59926

TABLE III
RESULTS FOR SET 10X100

#	ABHS		GR-ABHS		OL-ABHS	
	AVG	BEST	AVG	BEST	AVG	BEST
1	22474,82	22708	**22597,42**	22822	22595,50	22767
2	21920,08	22269	**22229,40**	22622	22216,86	22470
3	21356,08	21576	**21447,44**	21766	21435,98	21766
4	21986,14	22189	22114,96	22364	**22126,84**	22414
5	21769,74	22012	**21902,30**	22324	21881,68	22214
6	21723,76	22156	**21947,78**	22117	21935,82	22209
7	21117,88	21452	21394,40	21580	**21406,30**	21610
8	21693,82	21976	**21832,66**	22057	21826,14	22376
9	21542,70	22039	21699,30	21928	**21729,46**	21958
10	21441,50	21822	22069,70	22423	**22086,64**	22398
11	40550,70	41110	40662,40	41148	**40684,96**	40920
12	41509,04	41753	**41711,56**	42130	41668,04	41853
13	41611,72	41944	41824,78	42049	**41887,20**	42112
14	44655,74	45039	45074,62	45256	**45084,58**	45338
15	40935,38	41302	**41239,44**	41703	41221,28	41479
16	41962,84	42400	42292,44	42775	**42298,14**	42685
17	42691,78	43205	42903,62	43169	**42929,90**	43325
18	42291,72	42552	42418,32	42690	**42484,02**	42798
19	41425,60	41947	**41768,40**	41965	41751,28	42099
20	40244,34	40585	40655,82	40773	**40697,20**	41002
21	56908,94	57182	56951,66	57146	**56961,62**	57251
22	58378,34	58704	**58510,32**	58830	58409,62	58740
23	57825,14	58153	**58057,40**	58337	58000,24	58203
24	61382,18	61685	**61594,70**	61788	61526,38	61841
25	60451,48	60716	**60589,76**	60797	60583,14	60797
26	60840,90	61224	60801,54	61041	**60849,60**	61171
27	55829,48	56134	**55979,52**	56160	55966,70	56226
28	58908,24	59223	**58980,40**	59265	58969,84	59192
29	59668,78	59926	59747,98	59981	**59772,24**	60205
30	60267,78	60521	60278,00	60500	**60286,86**	60526

in 12 of the 30 instances.

In the set *10x100*, *OL-ABHS* and *GR-ABHS* obtained better average results in 15 of the 30 instances. On the other hand, *OL-ABHS* obtained best quality solutions in 17 of the 30 instances and, *GR-ABHS* in 13 of the 30 instances.

V. CONCLUSION

In this work we proposed the inclusion of Opposite Learning in *ABHS*, an Harmony Search approach for solving the well-known MKP. The opposite learning is obtained using three different strategies: *SOL*, *HOL* and *WOL*. Each strategy generates an antipheromone matrix that is used for the initialization of a third of the Harmony Memory in ABHS.

Results shows that the search process of *ABHS* is improved, in both instances sets by the inclusion of the opposite information. More specifically, considering both sets, *OL-ABHS* obtained better average results in 32 of 60 instances and best quality solutions in 39 of the 60 instances.

For future work, we are interested on including our Opposite Learning strategies in other metaheuristics for solving different optimization problems.

REFERENCES

[1] N. Rojas-Morales, M.-C. Riff, and E. Montero, "Ants can learn from the opposite," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) 2016*. ACM, 2016, pp. 389–396.

[2] N. Rojas-Morales, M.-C. Riff, and E. Montero, "Learning from the opposite: Strategies for ants that solve multidimensional knapsack problem," in *IEEE Congress on Evolutionary Computation, CEC 2016*, 2016, pp. 193–200.

[3] L. Wang, R. Yang, Y. Xu, Q. Niu, P. M. Pardalos, and M. Fei, "An improved adaptive binary Harmony Search algorithm," *Information Sciences*, vol. 232, pp. 58–87, 2013.

[4] J. Puchinger, G. R. Raidl, and U. Pferschy, "The multidimensional knapsack problem: Structure and algorithms," *Journal on Computing*, vol. 22, no. 2, pp. 250–265, 2010.

[5] A. R. Malisia, "Improving the exploration ability of ant-based algorithms," in *Oppositional Concepts in Computational Intelligence*. Springer Berlin Heidelberg, 2008, vol. 155, pp. 121–142.

[6] Q. Xu, L. Wang, N. Wang, X. Hei, and L. Zhao, "A review of opposition-based learning from 2005 to 2012," *Engineering Applications of Artificial Intelligence*, vol. 29, 2014.

[7] N. Rojas-Morales, M.-C. Riff Rojas, and E. Montero Ureta, "A survey and classification of opposition-based metaheuristics," *Computers & Industrial Engineering*, 2017.

[8] J. H. Drake, E. Özcan, and E. K. Burke, "A case study of controlling crossover in a selection hyper-heuristic framework using the multidimensional knapsack problem," *Evolutionary computation*, vol. 24, no. 1, pp. 113–141, 2016.

[9] I. Alaya, C. Solnon, and K. Ghédira, "Ant algorithm for the Multidimensional Knapsack Problem," in *International Conference on Bioinspired Optimization Methods and their Applications*, 2004, pp. 63–72.

[10] N. Rojas-Morales, M.-C. Riff, X. Bonnaire, and E. Montero, "New components with on-line control to improve harmony search," in *Computational Science and Engineering (CSE), 2013 IEEE 16th International Conference on*. IEEE, 2013, pp. 538–544.