Using Anti-pheromone to Identify Core Objects for Multidimensional Knapsack Problems: A Two-step Ants based Approach

Nicolás Rojas*
NicolasRojas@acm.org

Elizabeth Montero
EMontero@inf.utfsm.cl

María-Cristina Riff†
MCRiff@inf.utfsm.cl

Universidad Técnica Federico Santa María
Avenida España 1680
Valparaíso, Chile

ABSTRACT

This paper proposes a two-step ants algorithm for the Multidimensional Knapsack Problem. In the first step, the algorithm uses an Anti-pheromone to detect which objects are less suitable to be part of a near-optimal solution solving the opposite problem. From this information, in the second step an ant-based algorithm continues searching for better solutions trying to solve the real problem.

1. INTRODUCTION

Multidimensional Knapsack Problem (MKP) is defined as a knapsack with multiple resource constraints. It consists in selecting a subset of $n \leq N$ objects in such a way that the M knapsack constraints are satisfied maximizing the total profit. The constraints point out that the $Weight_{im}$ of selected objects don't exceeds the capacity b_m in each dimension m. The main objective of this paper is to answer the following question: Would it be possible to use Ant Colony Optimization (ACO) to help us to discard objects, and using this knowledge to solve Multidimensional Knapsack problems more efficiently? To answer this question, we introduce an Anti-pheromone structure [3] (also called Negative Pheromone [2]), and we compare it with the Ant Knapsack algorithm of Alaya et al [1] solving MKP. The contribution of this paper is to use the anti-pheromone idea to improve the Ant Knapsack (AK) algorithm [1]. AK is a MAX-MIN Ant System [5]. The pheromone is associated to promising pairs of objects. The heuristic knowledge of AK is based on the remaining capacity $RC_m = b_m - Weight_{jm}$, of adding the object j to the knapsack m. The idea here is to evaluate how much capacity is lost when a new object is included.

*Supported by CONICYT-PCHA/ National Doctorate / 2015-21150696

†Supported by Fondecyt Project no. 1151456

GECCO '15 July 11-15, 2015, Madrid, Spain

ACM ISBN 978-1-4503-3488-4/15/07.

DOI: http://dx.doi.org/10.1145/2739482.2764713

2. TWO-STEP ANTS ALGORITHM

We call our algorithm the Two-step Ants (TSA) and is based on the Ant Knapsack (AK) algorithm introduced in [1]. TSA has two main steps: the *First Step* searches for solutions that minimize the profit of objects. In other words, it detects worse solutions for the problem. The objective is to identify the arcs that belongs to these solutions and transmit that knowledge into the *Second Step*. The *Second Step* tries to solve the real problem maximizing the profit using AK. The experience obtained during the *First Step* defines which arcs will be un-desirable for ants during the *Second Step*. Algorithm 1 shows the structure of TSA.

Algorithm 1: `Two-Step Ants Algorithm`

```
1  InitializePheromoneTrails (τ_max);
2  Step ← First;
3  while Stop criterion is not met do
4      for ant k ← 1 to N_Total do
5          o_1 ← ChooseRandomlyObject ();
6          S_k ← {o_1};
7          UpdateRemainingCapacity (o_1,b);
8          CheckCandidates (Candidates,b);
9          while Candidates ≠ ∅ do
10             S_k ← ChooseAndAddNextObject(s);
11             UpdateRemainingCapacity (o_j,b);
12             CheckCandidates (Candidates,b);
13         end
14     end
15     EvaporateAnti-Pheromone (ρ) ;
16     DepositAnti-Pheromone (Step) ;
17     if TranslationTime then
18         Translate();
19         Step ← Second;
20     end
21 end
```

The *First Step* uses a defined number of iterations of the algorithm, then, the *Second Step* runs until the termination criterion is reached. As in AK, in *First Step* each ant starts its construction including a randomly chosen object in its solution, it modifies the remaining capacity and defines the *Candidates* list. Then, TSA decides which object o_j will be added to the solution, based in its profit and the remaining capacity (line 10). TSA uses a *Min-Min Strategy* that defines

the way the profit is considered in the step. The objective of this strategy is to minimize the evaluation function. Once the algorithm decides which object is going to be included, the strategy considers the profit of an object j as:

$$Profit_{conv}(j) = Max_{Profit} + Min_{Profit} - Profit_j \quad (1)$$

where Max_{Profit} is the largest profit of an object in the instance, Min_{Profit} the lowest one, $Profit_j$ is the profit of object j. The idea here is to solve the opposite problem detecting which pairs of objects are related in bad decisions.

To decide which object will be included next, we use the same probability presented for AK and the *Min-Min Strategy* will be considered in the heuristic knowledge η_S:

$$p_S(o_j) = \frac{(\tau_S(o_j))^\alpha * (\eta_S(o_j))^\beta}{\sum_{z \in Candidates}(\tau_S(o_z))^\alpha * (\eta_S(o_z))^\beta} \quad (2)$$

With a new object in the knapsack, the remaining capacities and the *Candidates* list are updated. Once each ant has constructed its solution, the algorithm updates the anti-pheromone matrix. Lines 15 and 16 show the evaporation and deposit of anti-pheromone (When *Step* is $First$). In this case, the worst quality ant of the current iteration S_{worst_it} will lay anti-pheromone. The amount of anti-pheromone that will be deposited is given by

$$q = 1/(1 + F(S_{worst\_it}) - F(S_{worst\_found}) \quad (3)$$

where F is the evaluation of the solution and $F(S_{worst\_found})$ the worst solution found so far.

When the *First Step* is finished, an Anti-pheromone matrix with information of pairs of objects that are part of lowest quality solutions is obtained. The main idea of using this information is to identify the *core* of objects for which it is hard to decide if they will be part of an optimal solution or not [4]. Hence, this information is translated to be used in the *Second Step* (line 18). The way the information is passed between the steps is called *translation rule*. To use it, we need to know the average of anti-pheromone in arcs at the end of the *First Step*. Then, if the pheromone in arc (o_i, o_j) is greater than the average, this arc will start the next step with τ_{min}. It decides considering the average in order to smooth the effect of intermediate bad decisions.

3. EXPERIMENTS

In this section we report the results obtained by the algorithms for solving MKP instances. The goal of these experiments is to evaluate the performance of using the anti-pheromone to obtain information about the *core* objects. To do this, we considered 10 instances created by Chu and Beasley. Each algorithm was executed 50 times during 2000 iterations. AK and TSA parameters values were set according to [1]: $\alpha = 1$, $\beta = 5$, $\rho = 0.01$, $N_{Total} = 30$, $\tau_{max} = 6$ and $\tau_{min} = 0.01$. TSA also requires to define how many iterations will be allocated to the *First Step* and how many to the *Second Step*. We defined 25% of the iterations for *First step* and 75% for *Second step*. Table 1 shows the average and the best solution reached. AK column shows the results presented in [1]. We can see that in 5 out of 10 cases, the average obtained by AK is better than TSA. However, TSA improves the average obtained in the other 5 cases. These results shows up a possible collaborative strategy, between TSA and AK, for solving these problems.

Instance	AK		TSA	
	Best	**Avg**	**Best**	**Avg**
5_100_1_0.25	24381	**24342**	24381	24338
5_100_2_0.25	24274	24247	24274	**24252**
5_100_3_0.25	23551	**23529**	23551	23528
5_100_4_0.25	23534	23462	23534	**23467**
5_100_10_0.25	24411	**24356**	24411	24344
10_100_1_0.25	**23064**	**23016**	23057	23009
10_100_2_0.25	22801	22714	22801	**22723**
10_100_3_0.25	22131	22034	22131	**22050**
10_100_4_0.25	22717	**22634**	**22772**	22611
10_100_10_0.25	22702	22591	22702	**22605**

Table 1: Comparison between AK and TSA

4. CONCLUSIONS

In this paper we proposed to solve Multidimensional Knapsack Problem using an Two-step ant based algorithm. The *First Step* is focused on the minimization problem in order to learn about non suitable objects. The information obtained from this step is given to the pheromone structure of the *Second Step*. We have evaluated our approach using a set of well-known instances from the literature. The results are very encouraging and show that learning about bad couples of objects can help the algorithm to identify the *core* of interesting objects. For future work, we will analyze the collaboration between anti-pheromone structure and another ant based algorithm. Our strategy is general, so we will evaluate it solving other problems.

5. ACKNOWLEDGMENTS

We would like to thank Christine Solnon for sending us the code for performing our experiments.

6. REFERENCES

[1] I. Alaya, C. Solnon, and K. Ghédira. Ant algorithm for the Multi-dimensional Knapsack Problem. In *International Conference on Bioinspired Optimization Methods and their Applications*, pages 63–72, 2004.

[2] A. Malisia and H. Tizhoosh. Applying Opposition-Based Ideas to the Ant Colony System. In *IEEE Swarm Intelligence Symposium*, pages 182–189, 2007.

[3] J. Montgomery and M. Randall. Anti-pheromone as a Tool for Better Exploration of Search Space. In *Ant Algorithms*, volume 2463 of *Lecture Notes in Computer Science*, pages 100–110. 2002.

[4] J. Puchinger, G. R. Raidl, and U. Pferschy. The Core Concept for the Multidimensional Knapsack Problem. In *Evolutionary Computation in Combinatorial Optimization*, pages 195–208. 2006.

[5] T. Stützle and H. H. Hoos. MAX–MIN Ant System. *Future generation computer systems*, 16(8):889–914, 2000.