A Cooperative Learning Approach for the Quadratic Knapsack Problem

Eduardo Lalla-Ruiz[1,4], Eduardo Segredo[2,3], and Stefan Voß[1(✉)]

[1] Institute of Information Systems, University of Hamburg, Hamburg, Germany
{eduardo.lalla-ruiz,stefan.voss}@uni-hamburg.de
[2] School of Computing, Edinburgh Napier University, Edinburgh, UK
e.segredo@napier.ac.uk
[3] Dpto. de Ingeniería Informática y de Sistemas, Universidad de La Laguna, San Cristóbal de La Laguna, Spain
esegredo@ull.edu.es
[4] Department of Industrial Engineering and Business Information Systems, University of Twente, 7522NB Enschede, The Netherlands
e.a.lalla@utwente.nl

Abstract. The *Quadratic Knapsack Problem* (QKP) is a well-known optimization problem aimed to maximize a quadratic objective function subject to linear capacity constraints. It has several applications in different fields such as telecommunications, graph theory, logistics, hydrology and data allocation, among others. In this paper, we propose the application of a novel population-based metaheuristic referred to as Multi-leader Migrating Birds Optimization (MMBO), which exploits the concepts of cooperation and communication along the search leading to a collective learning, to solve a wide range of well-known QKP instances.

1 Introduction

The *Quadratic Knapsack Problem* (QKP) is a knapsack problem introduced by Gallo *et al.* [5] that includes the relationship among the items within a quadratic objective function. Formally, we are given a set of items $N = \{1, ..., n\}$, where each item i has a given weight $w_i > 0$, and an $n \times n$ profit matrix B with entries $b_{i,j}$ that indicates the benefit obtained when an item is packed with respect to itself and other items. In other words, if an item i is selected, then the profit obtained is equal to $b_{ii} + \sum_{j \in N \setminus \{i\}} b_{ij}$. Note that matrix B is symmetric, i.e., $b_{ij} = b_{ji}$. The goal of the QKP is thus determining the items to be packed in the knapsack taking into account the weight capacity c such that the total profit of the items packed is maximized. The selection of the items is ruled by the binary variable x_i which is equal to 1 if item i is selected and 0 otherwise. The formal definition of the QKP is as follows.

$$\max z(QKP) = \sum_{i \in N} \sum_{j \in N} b_{ij} x_i x_j \tag{1}$$

R. Battiti et al. (Eds.): LION 12 2018, LNCS 11353, pp. 31–35, 2019.
https://doi.org/10.1007/978-3-030-05348-2_3

$$\sum_{i \in N} w_i x_i \leq c \tag{2}$$

$$x_i \in \{0, 1\}, \quad i \in N \tag{3}$$

Note that, if $b_{ij} = 0$ for $i \neq j$ then the QKP can be reduced to the *Knapsack Problem*. Moreover, the *Clique Problem* can be treated as a particular case of the QKP, where the *Max Clique* can be solved by a QKP algorithm by using a binary search [2].

In this work, we propose the application of a novel population-based metaheuristic, called *Multi-leader Migrating Birds Optimization* (MMBO) [6]. This approach exploits the communication among a population of individuals during the search, thus enabling cooperative learning. We test our method for the latest benchmark suite proposed for the QKP [3]. We attain competitive results, in terms of the objective function value obtained at the end of the runs in comparison to the best-performing state-of-the-art approaches used in those problem instances.

2 Multi-leader Migrating Birds Optimization

For solving the QKP, we propose the use of MMBO (Algorithm 1), which is a decentralized cooperative search approach inspired by the flight formation of migratory birds. In MMBO, the population is denoted as $P = \{1, 2, ..., p\}$, where p is the number of individuals representing solutions of the optimization problem at hand. During the search, individuals are distributed in a line formation, i.e., $(1, 2, ..., p)$, where individual 1 is directly connected to individual 2, and individual 2 is connected to individuals 1 and 3, and so on. Based on that line formation, a relationship structure is established according to a given *relationship criterion*, for instance, in terms of the objective function value associated with each member of the population. By means of that criterion, the role of each pair of individuals is determined, i.e., which individual provides and which individual receives information during the search.

Starting from each individual in P, k feasible neighbors are generated through a predefined neighborhood structure. In this work, two decision variables of a given individual are uniformly selected at random and their corresponding binary values are flipped in order to produce a novel neighbor. Depending on the relationship criterion and how information is shared among individuals, different roles arise:

– **Leader**. It is that individual with the best objective function value when compared to its adjacent individuals. Hence, it does not receive information from any individual, but shares it, in the form of δ neighbors, with its adjacent individuals. Each leader generates k neighbors. The set of leaders is denoted as P_L.

– **Follower**. It is that individual which explores the search space considering its own information and the information received from the individuals in front of it within the relationship structure. It generates $k - \delta$ neighbors and receives δ neighbors. The set of followers is denoted as P_F.
– **Independent**. It is that individual which is not included into any of the above categories because it has associated the same objective function value than its adjacent individuals. Hence, it does not exchange information with any other individual, but generates k neighbors. The set of independent individuals is denoted as P_I.

Algorithm 1 Pseudocode of MMBO

Require: p, K, k, and δ
Initialize the population P, which consists of p individuals generated at random
while (K neighbors have not been generated) **do**
  Determine the interaction among the individuals of P and establish the relationship structure
  while (the stopping criterion associated with the relationship structure is not met) **do**
    Generate k neighbors starting from each individual included within $P_L \cup P_I$
    Replace each individual included within $P_L \cup P_I$ by its fittest neighbor if the latter is fitter than the former
    Replace each individual included within P_I by its fittest neighbor
    for all (individual $f \in P_F$) **do**
      Generate $k - \delta$ neighbors starting from f
      Get the best unused δ neighbors from the previous individuals of f in the relationship structure
      Replace f by its fittest neighbor if the latter improves the former
    end for
  end while
end while
Return the fittest individual in P

3 Numerical Results

The proposed approach has been implemented in Java and executed on a computer equipped with an Intel i7 CPU 3.5 GHz and 16 GB of RAM. By preliminary experiments, we identified the following parameters $p = 20$, $\delta = 1$, and $k = 5$ with a stopping criterion $K = |n|^2$ neighbors. The problem instances used are those proposed in [3] by following the guidelines of previous works. The instances are generated for different density values, i.e., different percentage of non-zero cross benefits in $B = \{b_{ij} : i, j \in N, i < j\}$, where each b_{ij} is chosen from the interval $[1, 100]$. The knapsack capacity c is selected from $[50, \sum_{j=1}^{n} w_j]$.

Table 1 shows the computational results for instances with $n = 100$ items, which were generated by considering different density values (dst). The methods selected for comparison purposes are the best-performing ones in the related literature for the instances proposed in [3]. They are based on a *Dynamic Programming Heuristic* (DPH) and a *Non-Delayed Relax-and-Cut* (CSL) [4]. We should note that execution times are measured as integer values in [3]. Hence, some execution times are reported as 0 when the time invested is lower than 1 second. Therefore, we report the upper bound of the said times to avoid those cases. Our approach is able to reduce considerably the execution time with respect to the

best approach in terms of the objective function value (CSL) while reporting a new best solution considering one of the instances ($id = 4$). Finally, it is worth mentioning that MMBO is able to reach the optimal solution in those problem instances proposed in [1] and solved to optimality.

4 Conclusions

In this work, we have proposed a MMBO approach for the Quadratic Knapsack Problem. Due to its self-organization and cooperation dynamics, it allows individuals to learn along the search process in a collective way. The collaborative relationship structure enhances the diversification of the search as individuals can be distributed over the search space. Intensification is addressed by the sum of efforts of the individuals belonging to the same group located in a particular region of the search space. Our algorithm reports high-quality solutions in terms of the objective function value while investing shorter computational times compared to state-of-the-art approaches. In this regard, we even attained a new best solution for one of the instances tested.

Table 1. Numerical results for instances of $n = 100$ items. Bold represents a new best solution

Instance			CSL		DPH		MMBO				
n	dst	id	Obj	Time	Obj	Time	Max	Avg.	Min.	σ^2	Time
100	25	1	53774	3	53757	2	53774	53759.6	53723	371.64	0.08
		2	7082	5	7076	1	7082	7079.6	7076	8.64	0.06
		3	60875	2	60875	1	60875	60875	60875	0.00	0.09
		4	18386	6	18386	2	18386	18273.4	18224	2651.24	0.04
		5	43014	4	43014	1	43014	43002.2	42896	1253.16	0.08
		6	50484	4	50484	1	50484	50479.6	50473	29.04	0.06
		7	21769	4	21769	2	21769	21755.4	21701	739.84	0.07
		8	30687	5	30687	1	30687	30652.4	30555	2724.84	0.06
		9	28719	6	28719	1	28719	28666.4	28585	2037.44	0.13
		10	5463	4	5421	2	5463	5461	5443	36.00	0.08
100	50	1	34653	5	34653	1	34653	34653	34653	0.00	0.08
		2	43178	5	43169	1	43178	43170.9	43156	70.69	0.06
		3	46243	6	46243	2	46243	46232.5	46138	992.25	0.09
		4	48894	5	48992	1	**49030**	49004.7	48894	1666.41	0.04
		5	41515	6	41515	1	41515	41515	41515	0.00	0.08
		6	71982	4	71982	2	71982	71982	71982	0.00	0.06
		7	69146	6	69177	1	69177	69091.8	68985	7414.56	0.07
		8	83085	3	83085	1	83085	83057.5	82948	2045.85	0.06
		9	9772	4	9772	2	9772	9772	9772	0.00	0.13
		10	62465	6	62407	1	62465	62415	62081	12918.4	0.08
Average			41559.30	4.65	41559.308	1.35	41567.65	41544.95	41483.75	1748	0.07

As future work, we aim to extend the results provided herein to analyze the different features provided by the approach to solve the QKP in more detail.

References

1. Billionnet, A., Soutif, É.: An exact method based on lagrangian decomposition for the 0–1 quadratic knapsack problem. Eur. J. Oper. Res. **157**(3), 565–575 (2004)
2. Caprara, A., Pisinger, D., Toth, P.: Exact solution of the quadratic knapsack problem. INFORMS J. Comput. **11**(2), 125–137 (1999)
3. Cunha, J.O., Simonetti, L., Lucena, A.: Lagrangian heuristics for the quadratic knapsack problem. Comput. Optim. Appl. **63**, 97–120 (2016)
4. Fomeni, F., Letchford, A.: A dynamic programming heuristic for the quadratic knapsack problem. INFORMS J. Comput. **26**(1), 173–182 (2013)
5. Gallo, G., Hammer, P.L., Simeone, B.: Quadratic knapsack problems. In: Padberg, M.W. (ed.) Combinatorial Optimization, pp. 132–149. Springer, Berlin (1980). https://doi.org/10.1007/BFb0120892
6. Lalla-Ruiz, E., de Armas, J., Expósito-Izquierdo, C., Melián-Batista, B., Moreno-Vega, J.M.: Multi-leader migrating birds optimisation: a novel nature-inspired metaheuristic for combinatorial problems. Int. J. Bio-Inspired Comput. **10**, 89–98 (2017)