Optimization of the p-Hub Median Problem via Artificial Immune Systems

Stephanie Alvarez Fernandez[1(✉)], Gabriel Lins e Nobrega[2], and Daniel G. Silva[1]

[1] Department of Electrical Engineering, University of Brasília, Brasilia, Brazil
stephaniemaf@gmail.com, danielgs@ene.unb.br
[2] Exact Sciences Institute, University of Brasília, Brasilia, Brazil
gabriel.lins97@gmail.com

Abstract. Recent advances in logistics, transportation and in telecommunications offer great opportunities to citizens and organizations in a globally-connected world, but they also arise a vast number of complex challenges that decision makers must face. In this context, a popular optimization problem with practical applications to the design of hub-and-spoke networks is analyzed: the Uncapacitated Single Allocation p-Hub Median Problem (USApHMP) where a fixed number of hubs have unlimited capacity, each non-hub node is allocated to a single hub and the number of hubs is known in advance. An immune inspired metaheuristic is proposed to solve the problem in deterministic scenarios. In order to show its efficiency, a series of computational tests are carried out using small and large size instances from the Australian Post dataset with node sizes up to 200. The results contribute to a deeper understanding of the effectiveness of the employed metaheuristic for solving the USApHMP in small and large networks.

Keywords: Artificial immune systems · p-Hub Median Problem · Metaheuristics · CLONALG

1 Introduction

The design of hub-and-spoke networks addresses decisions about where to allocate or install hubs (facilities), considering customers' or users' demands that must be served, in order to optimize one or more criteria. The "hub" term is used to refer to schools, factories, warehouses, telecommunication antennas, etc., while "customers" refer to neighborhoods, sales units, students, etc. Localization problems are combinatorial by nature, since they consist of selecting from a discrete, finite set of data the best subset that satisfies certain criteria. Many are highly complex and costly from a computational point of view.

There are many variants of the hub localization problem (HLP) [25]: the p-hub median problem, the p-hub center problem, the capacitated/uncapacitated hub location problem, and the hub covering problem. Moreover, HLP may

C. Paternina-Arboleda and S. Voß (Eds.): ICCL 2019, LNCS 11756, pp. 350–362, 2019.
https://doi.org/10.1007/978-3-030-31140-7_22

be classified by the way in which the requested points are assigned or allocated to hubs. In this sense, they can assume one of the two allocation schemes: *(i)* single allocation scheme, where each node must be assigned to exactly one hub node (*i.e.*, all flows from/to each node go only via an assigned hub); and *(ii)* multiple allocation scheme, where nodes are allowed to communicate with more than one hub. Furthermore, different constraints may arise, including capacity restrictions on the traffic volume a hub can concentrate. A straightforward version of this problem is the uncapacitated single allocation p-hub median problem (USApHMP), which assumes that the capacity of each hub is virtually unlimited or, at least, far beyond the expected demand. In these types of configurations, the hubs serve as connection points between two installations, allowing to replace a large amount of direct connections between all pairs of nodes, hence, minimizing the total transportation cost of the network.

Recently, the application of heuristic methods to HLP has received considerable attention by many researchers. On one hand, metaheuristics based on General Variable Neighborhood Search [15], GRASP [26], Scatter Search [4,21], Iterated Local Search [3], among others [16,29,31] have been proposed. Furthermore, recent successful approaches using exact methods are [22,23]. On the other hand, bio-inspired methodologies such as GA have been also proposed in order to solve other variants of the problem such as the uncapacitated single allocation hub location problem, in which the number of hubs is not given in advance [13], or the uncapacitated multiple allocation p-hub median problem, in which one node can be allocated to more than one hub [24]. Furthermore, the application of Artificial Immune Systems (AIS) methodology such as Clonal Selection for solving capacitated HLP and p-hub median problems have been applied [19,20].

Particularly, AIS have been widely adopted for optimization purposes. One can find different approaches of AIS algorithms for solving problems such as optimization of distributed generation in distribution systems [7], optimization of material handling system [18], constrained optimization problems [33], dynamic constrained optimization [34], reverse logistics [10], design problems of energy suply networks [32], distributed wireless networks [28], distribution scheduling problem [27] and many others. The reader is referred to [6] for a recent survey of the application of AIS in optimization.

Regarding the application of bio-inspired metaheuristics for specifically solving the USApHMP, just a few articles were found in the literature. In [17], the authors solved this problem by using two distinct GA approaches. They presented solutions with the Civil Aeronautic Board (CAB) and Australian Post (AP) datasets. Likewise, in [14], the authors considered an approach consisting on the application of Clonal Selection for solving the same problem, however, there is no experimental results to support this outlook.

By relying on the aforementioned explanations, the CLONALG method is proposed to solve the USApHMP. In contrast with a standard GA and other bio-inspired proposals which have been applied to solve other variants of the problem, immune inspired algorithms such as CLONALG present an intrinsic capacity of maintaining diversity among the candidate solutions during the execution, which

can be decisive to increase the probability that the global optimum or a good local optimum can be reached. Although it is known that there is no particular global search strategy with superior performance for a wide class of problems, these arguments support the application of such metaheuristic in the context of USApHMP.

The rest of the paper is organized as follows: Sect. 2 describes the mathematical model of the USApHMP. The proposed methodology is presented in Sect. 3. The experiments and results are provided in Sect. 4. Finally, concluding remarks on this work are provided in Sect. 5.

2 Formal Problem Description

The USApHMP consists of choosing p nodes from a given network of n nodes as hubs, while allocating the remaining $(n-p)$ nodes to them, in order to minimize the total transportation cost over the network. As mentioned before, this problem differs from other HLPs because the number of p hubs is given beforehand. We use the integer programming formulation given in [1,17] which, in turn, is based on the very first model proposed by O'kelly [25] and has been extensively applied for solving small to large instances of the problem. Accordingly, the following notation and formulation is given.

Notation:

- $N = \{1, 2, \cdots, n-1, n\}$: set of n distinct nodes in the network. Each node refers to origin/destination (O/D) points or potential hub location;
- C_{ij}: cost per unit flow from each origin node i to destination node j;
- W_{ij}: amount of flow from i to j;
- $X_{ik} = \begin{cases} 1, & \text{if non-hub node } i \text{ is allocated to a hub node } k \\ 0, & \text{otherwise} \end{cases}$
- $X_{kk} = 1$: implies that the node k is a hub;
- $\mathcal{X}$: cost for collecting flow from the origin non-hub node to its hub;
- τ: cost of transferring the collected flow between the interconnected hubs;
- δ: cost of distributing the flow from the hub to the destination node;

Given the aforementioned notation the problem is formulated as follows:

$$\text{Minimize:} \quad \sum_{i,j,k,l \in N} W_{ij}(\mathcal{X} C_{ik} X_{ik} + \tau C_{kl} X_{ik} X_{jl} + \delta C_{jl} X_{jl}) \tag{1a}$$

$$\text{Subject to:} \quad \sum_{k=1}^{n} X_{kk} = p, \tag{1b}$$

$$\sum_{k=1}^{n} X_{ik} = 1, \forall\ i \in N \tag{1c}$$

$$X_{ik} \leq X_{kk} \quad \forall\ i \in N \tag{1d}$$

$$X_{ik} \in \{0, 1\} \quad \forall\ i \in N \tag{1e}$$

Equation 1b gives the exact number of hubs to be selected from the node network. Equation 1c is the single allocation constraint, ensuring that the flow from any origin non-hub node i is sent through one and only one hub node to which the node i is allocated. Equation 1d guarantees that a non-hub node is able to be allocated to only one hub, thus avoiding direct movement between O/D nodes.

3 Methodology

In this section, primarily based on [1], the key principles of AIS and CLONALG method are derived. AIS are a class of algorithms inspired by the mammalian's immune system. In the context of engineering and computing, the application of the immunological principles of decentralization and diversity maintenance are often useful in solving problems with large solutions spaces, such as the USApHMP.

The immune system characteristics have received attention from the research community in order to design new intelligent frameworks. Properties such as: automatic recognition of antigen's characteristics, pattern recognition and memorization capabilities, self-organizing memory, adaptation ability, immune response adaptation, learning from examples, distributed and parallel data processing, multilayer structure and generalization capability [11] have been reproduced for solving optimization problems, like the one of this work, as well as identification of non-linear systems [2,30] and machine learning related problems [5].

In the context of USApHMP, the cells and their antibodies are analogous to a candidate solution for the optimization problem posed by 1a. Thus, it is possible to immunologically simulate such problem, seeking solutions effectively. With this in mind, an immune-inspired metaheuristic is proposed for solving the USApHMP: the CLONALG algorithm, a classical technique that emulates the clonal selection principle for optimization tasks.

3.1 CLONALG Algorithm

Clonal selection algorithms are inspired by the clonal selection theory, which basically addresses the way the immune system responds to infectious agents. It is result of the work accomplished by [8], which in turn, served as inspiration for CLONALG [9], a popular AIS that emulates the cloning and hypermutation process, key principles of the original theory. The improvement of the candidate solutions pool, in CLONALG, follows the steps of all clonal selection-based algorithm: clone, mutate, select and replace.

The clonal selection theory argues that, when an antigen penetrates the host body, a selection of lymphocytes capable of recognizing it takes place. Lymphocytes with higher antigen reactivity would be selected for clone expansion over those with lower neutralizing capability. Each clone should have a unique antigen receiver, *i.e.* it should not be present in any other. The idea is that only cells capable of properly recognizing antigens should survive and generate offspring, *i.e.* the process promotes a maturation of the immune system response to the infection.

Algorithm 1. CLONALG(nodes, pHubs, β, ρ, Ninitial, nC, b)

```
Ab ← random(Ninitial)
while Stop condition is not met do
    Solve fit ← affinity(Ab)
    C ← clone(Ab, nC, β)
    C* ← mutate(C, fit, ρ)
    Fit' ← affinity(C*)
    R ← select(C*, Fit')
    Ab ← replace(R, random(b))
end
return Ab
```

Based on the clonal selection principle, the CLONALG has been proposed as a method to be applied both in optimization, as presented in this work, and in machine learning. CLONALG is largely referenced by the study presented in [9], in which the sensitivity to its parameters and its performance in a set of test problems were analyzed. Its pseudocode for solving the USApHMP, based on the description presented in [1], can be found in Algorithm 1. The input parameters are: (i) the number of nodes as well as the number of hubs given by the problem; (ii) the clonal factor β; (iii) parameter ρ which controls the shape of the mutation rate; (iv) the size $Ninitial$ of the antibody[1] pool Ab; (v) the number of clones nC; and (vi) the parameter b for selecting the number of new antibodies that will replace the lowest affinity ones from Ab.

The procedure starts generating a pool of antibodies with fixed size $Ninitial$. The authors of [9] proposed that the generation of those antibodies occurs randomly for a greater diversity within the population. In the case of the USApHMP, every member of Ab represents a solution for the problem, *i.e.* an allocation matrix X. First, the given number of p hubs are randomly chosen following a uniform probability distribution, then, the remaining nodes are allocated to their nearest hubs. Next, every antibody in Ab is evaluated by the affinity/fitness function

$$f^{Ag}(Ab_i), \tag{2}$$

where Ag represents the antigens, and the fitness is the cost function given by Eq. 1a. This way, the Ab solutions with lower fitness also have lower cost, therefore, are the best ones. Following, the amount of clones nC to be generated for each individual is calculated as defined in Eq. 3:

$$nC = \text{round}(\beta \cdot Ninitial), \tag{3}$$

where β is the clonal factor. Then, the affinity maturation process is applied at an α rate proportional to the affinity obtained from Eq. 2, as defined in the following equation:

[1] Unlike in nature, there is no distinction for the terms antibodies/lymphocytes/cells in the context of AISs.

Algorithm 2. mutate(Ab, pHubs)

```
while Stop condition is not met do
    select mode pHubs or nodes
    if pHubs then
        oldHub ← select one hub randomly
        newHub ← select one node randomly that is not a hub
        replace oldHub with newHub
        disconnect all nodes
        reconnect all nodes to closest hub
    else
        node ← select one node randomly
        disconnect node from its corresponding hub
        connect node to a new randomly selected hub
    end
end
return Ab*
```

$$\alpha = \rho \cdot fit, \tag{4}$$

with ρ being the normalized affinity in $[0, 1]$ of the correspondent antibody. Differently from the proposal in [9], originally designed for function maximization, note that the mutation rate is proportional to their parent's affinity, *i.e.*, the greater the affinity (the USApHMP cost), the greater the mutation intensity.

Afterwards, the affinity of every clone is calculated and it is selected the best one among the clones and their parents to be kept. Finally, the main loop is concluded where the b worse antibodies are replaced with new randomly generated individuals. The process repeats itself until the stopping criteria is met.

The CLONALG mutation operator is a crucial aspect of the method that directly affects the diversity of the candidate solutions and the exploration of the search space. Recall that an antibody represents an allocation matrix of the hubs and nodes, *i.e.* the matrix X in Eq. 1a, then Algorithm 2 presents the pseudo-code for this operator. The procedure selects between $pHubs$ or $node$ mode based on the mutation rate α. If the procedure $pHubs$ is selected, one of the Ab hubs is chosen in order to be replaced with another node which becomes a hub, such node is randomly selected. Then, it reconnects all nodes to their current closest hub. If the $node$ procedure is selected, instead of randomly choosing a hub, a node is selected and its current hub is changed and replaced from among Ab hubs possibilities. By choosing either option, one can expect that candidate solutions are able to escape out of local minima or explore local regions.

Once the set of clones have been generated, each clone is then mutated and an affinity-based selection among the clones and their parent ensures that the solutions generated after the clone step have, on average, higher affinities than those of the early primary response. It is worth stressing that the mutation operator follows the restrictions imposed by the problem formulation, seen in Eq. 1a.

In general, the performance of AIS can be affected by variations on its parameters. CLONALG, particularly, have crucial parameters that control the algorithm exploration capacity: (*i*) the antibody population size; (*ii*) the number of clones; (*iii*) the remainder replacement size; and (*iv*) the stopping criterion. Understanding the effects of each parameter is crucial for a proper understanding of the algorithm as well as for its refinement. The impact that causes changing the setting of each parameter is summarized in Table 1, as presented in [1].

Table 1. CLONALG parameters

Operator	Function
Antibody population size, *Ninitial*	If excessive, an overloaded number of redundant antibodies can be generated
Number of clones, nC	Excessive number of clones may lead to redundancy in the search process, while a few number of clones may cause ineffective results in the search for a favorable mutation
Remainder replacement size, b	An excessive addition of antibodies reduces the convergence speed but provides a higher search space exploration; on the other hand, adding few antibodies may impair the search for a global optimum
Stopping criterion, *maxIt*	A great number of iterations impacts the execution time. On the other hand, few iterations may lead to poor performance of the algorithm

4 Experiments and Results

In this section, some numerical experiments are considered to check the performance of the proposed CLONALG algorithm to solve the USApHMP. For the sake of examination, the AP dataset is used, its details are described in [12]. All computational experiments were performed on an Intel® i7 7700HQ v4 at 2.8 GHz running the Arch Linux 4.19.4 operating system. The method has been implemented as C application.

As previously mentioned in Sect. 3, the CLONALG parameters adjustment is crucial for a proper performance of the method. Hence, they were defined via a preliminary parameter-selection process, which comprised 10 trials of the algorithm for each possible configuration of $Ninitial \in \{1, 3, 5, 7, 9\}$, $nC \in \{5, 10, 20, 30\}$, $b \in \{0.1, 0.3, 0.5, 0.7, 1\}$ and $maxIt \in \{300, 500, 700, 1000, 1500\}$. In order to minimize runtime and the final cost for each instance of the problem, the trials were executed with two n, p representative scenarios of USApHMP. For instances with "Small" size, those with n up to 50, trials were executed with $n = 10$ and $p = 5$. For instances with "Large" size, those with n ranging from 100 to 200, trial configuration was such that $n = 100$ and $p = 10$. Parameters that showed promising results regarding execution time and cost were chosen and are displayed in Table 2. The algorithm stops if it global solution (when previously known) for that particular instance, if the best solution in the population

remains without changes through 50 iterations for small instances and 200 for large ones, or if $maxIt$ iterations is achieved.

Table 2. CLONALG parameters

Parameter	Small	Large
$Ninitial$	3	7
nC	5	20
b	1	1
$maxIt$	300	1000

After setting each parameter values, we have run the algorithm 20 times, in order to collect the information regarding its performance for solving the different USApHMP instances. The quality of each solution is evaluated as a percentage gap between the solution found by CLONALG and the solutions presented in [3] and [17] using

$$Gap\ (Cost, Cost_{BKS}) = 100 \times \left(\frac{Cost - Cost_{BKS}}{Cost_{BKS}} \right). \tag{5}$$

In addition, runtime information has been used to compare the computational burden of the aforementioned methods.

Following, results obtained for the set of instances are summarized in Table 3. The first two columns, "n" and "p" indicate the number of nodes and hubs, respectively. The following two columns "*Cost(1)*" and "*Time(s)*" give the best solutions obtained by BRILS, the metaheuristic proposed by the authors in [3] and which, to the best of our knowledge, provides the best average runtime found in literature regarding the solution of the USApHMP using the AP dataset. The optimal solutions are highlighted in bold, notice that the optimal solutions reported in [3] match with the optimal solutions found by CPLEX solver. The next two columns denoted by "*Cost(2)*" and "*Time(s)*" present the best solution and time reported by the GA method in [17]. This classic technique was chosen for comparison because it is, like CLONALG method, a bio-inspired, population-based, non-deterministic algorithm. Following, the next two columns denoted by "*Cost(3)*" and "*Time(s)*" present the best solution and time obtained when applying our proposed CLONALG algorithm. Finally, the last two columns present the percentage gaps regarding the solutions found by CLONALG and the solutions reported when using BRILS and the GA method, respectively.

By looking at the results presented, we can conclude that the proposed AIS can operate efficiently for solving the presented problem in most of the instances in a very competitive time, the average runtime when applying CLONALG is 7.21 s, which is about 52% of the average time consumed by its similar counterpart, the population-based GA technique [17]. Furthermore, it follows that CLONALG solutions outperform the results reported by the genetic method.

Table 3. Solutions found by CLONALG for the AP instances

Instance		BRILS [3]		GA [17]		CLONALG '19		GAP	
n	*p*	*Cost(1)*	*Time(s)*	*Cost(2)*	*Time(s)*	*Cost(3)*	*Time(s)*	*Cost(3)-(1)*	*Cost(3)-(2)*
10	2	**167493.08**	0.00	167493.06	0.00	167493.065	0.00	0.00	0.00
	3	**136008.14**	0.01	-	-	136008.126	0.00	0.00	-
	4	**112396.08**	0.01	112396.07	0.02	112396.068	0.00	0.00	0.00
	5	**91105.38**	0.01	91105.37	0.03	91105.371	0.00	0.00	0.00
20	2	**172816.69**	0.02	172816.69	0.01	172816.690	0.00	0.00	0.00
	3	**151533.08**	0.02	151533.08	0.02	151533.084	0.00	0.00	0.00
	4	**135624.88**	0.03	135624.88	0.03	135624.884	0.00	0.00	0.00
	5	**123130.09**	0.03	123130.09	0.06	123130.095	0.00	0.00	0.00
25	2	**175541.98**	0.03	175541.98	0.03	175541.978	0.00	0.00	0.00
	3	**155256.32**	0.03	155256.32	0.04	155256.323	0.00	0.00	0.00
	4	**139197.17**	0.03	139197.17	0.07	139197.169	0.00	0.00	0.00
	5	**123574.29**	0.03	123574.29	0.03	123574.289	0.00	0.00	0.00
40	2	**177471.68**	0.04	177471.67	0.05	177471.674	0.00	0.00	0.00
	3	**158830.55**	0.04	158830.54	0.09	158830.545	0.01	0.00	0.00
	4	**143968.88**	0.05	143968.88	0.05	143968.876	0.01	0.00	0.00
	5	**134264.97**	0.05	134264.97	0.18	134264.967	0.01	0.00	0.00
50	2	**178484.29**	0.05	178484.29	0.14	178484.286	0.01	0.00	0.00
	3	**158569.94**	0.05	158569.93	0.13	158569.933	0.02	0.00	0.00
	4	**143378.05**	0.08	143378.05	0.20	143378.046	0.03	0.00	0.00
	5	**132366.96**	0.06	132366.95	0.23	132366.953	0.04	0.00	0.00
100	5	136984.7	0.88	136929.44	2.15	136929.44	0.42	−0.04	0.00
	10	106469.57	0.50	106829.15	11.15	106469.57	4.01	0.00	−0.34
	15	90534.79	0.46	90534.78	13.18	90533.52	7.97	0.00	0.00
	20	80305.1	3.86	80471.84	34.36	80270.96	9.79	−0.04	−0.25
200	5	140062.65	1.08	140450.08	18.34	140062.65	15.68	0.00	−0.28
	10	110147.66	0.39	110648.72	57.34	110147.66	29.84	0.00	−0.45
	15	94695.79	0.87	95857.69	81.92	94495.06	59.18	−0.21	−1.42
	20	85006.05	9.11	86069.21	151.20	85337.42	74.71	0.39	−0.85
Average			0.64		13.74		7.21		

Moreover, from the results in Table 3, one can see that three solutions obtained by CLONALG are better and only one case is worse than the respective ones obtained by BRILS [3]. In average terms, it can be concluded that most of the best solutions obtained by GA and BRILS regarding large instances ($n = \{100, 200\}$) are improved by CLONALG.

In order to better depict the CLONALG performance, Fig. 1 shows the fitness behavior among the population through the generations. The figure was obtained when running the algorithm for solving the problem with $n = 50$ nodes and $p = 5$ hubs. Observe that the method can provide in few iterations solutions with low gap to the optimal value.

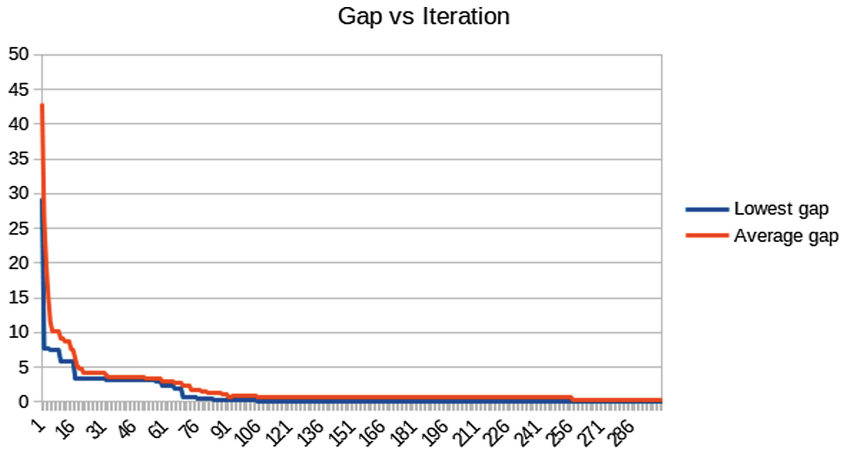

Fig. 1. Simulation results for the network with $n = 50$ nodes and $p = 5$ hubs.

5 Concluding Remarks

In this work we have proposed an Artificial Immune System, the CLONALG optimization algorithm, for solving the USApHMP. The most popular benchmark problem found in the literature, the AP dataset, was considered. A series of tests were carried out, which highlighted in particular the performance of the CLONALG algorithm in the 28 instances that were evaluated. The overall results indicated that the method could outperform its counterpart, the population-based GA metaheuristic, both in terms of quality of solutions and time performance. Furthermore, the algorithm could obtain better solutions than the fast metaheuristic BRILS in 3 out of 4 large instances.

One can conclude that the findings presented in this work add to a growing body of literature on the USApHMP, as they reinforce the effectiveness of AIS as global optimization technique and indicate potential applications of the same method to variations of the USApHMP, where situations with capacity constraints, multiple hub allocation or undefined number of hubs might yield optimization spaces with multiple optima, which positively accounts for the intrinsic population diversity capability of AIS methods.

There are several perspectives for future studies, such as the extension of the experiments for scenarios with more nodes and hubs and a sensitivity analysis of CLONALG with respect to its parameters, which unfortunately were not possible to perform in time for this work. Additionally, studying the potential association between BRILS and CLONALG in order to achieve high quality and, simultaneously, low runtime solutions can lead to interesting new results.

References

1. Alvarez Fernandez, S.: A metaheuristic and simheuristic approach for the p-hub median problem from a telecommunication perspective. Ph.D. thesis, University of Brasília (2018). http://hdl.handle.net/10803/666752
2. Alvarez Fernandez, S., Fantinato, D.G., Montalvao, J., Attux, R., Silva, D.G.: Immune-inspired optimization with autocorrentropy function for blind inversion of Wiener systems. In: 2018 IEEE Congress on Evolutionary Computation (CEC), pp. 1–7. IEEE (2018)
3. Alvarez Fernandez, S., Ferone, D., Juan, A.A., Silva, D., Armas, J.: A 2-stage biased-randomized iterated local search for the uncapacitated single-allocation p-hub median problem. Trans. Emerging Telecommun. Technol. **29**(9), e3418 (2018)
4. Amin-Naseri, M.R., Yazdekhasti, A., Salmasnia, A.: Robust bi-objective optimization of uncapacitated single allocation p-hub median problem using a hybrid heuristic algorithm. Neural Comput. Appl. **29**(9), 511–532 (2018)
5. Aydin, I., Karakose, M., Akin, E.: A multi-objective artificial immune algorithm for parameter optimization in support vector machine. Appl. Soft Comput. **11**(1), 120–129 (2011)
6. Bernardino, H.S., Barbosa, H.J.C.: Artificial immune systems for optimization. In: Chiong, R. (ed.) Nature-Inspired Algorithms for Optimization, pp. 389–411. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-00267-0_14
7. Bhadoria, V.S., Pal, N.S., Shrivastava, V.: Artificial immune system based approach for size and location optimization of distributed generation in distribution system. Int. J. Syst. Assur. Eng. Manag. **10**(3), 339–349 (2019)
8. Burnet, F.M.: Clonal Selection and After. In: Bell, G.I., Perelson, A.S., Pimbley Jr., G.H. (eds.) Theoretical Immunology, pp. 63–85. Marcel Dekker Inc., New York (1978)
9. de Castro, L.N., Von Zuben, F.J.: Learning and optimization using the clonal selection principle. IEEE Trans. Evol. Comput. **6**(3), 239–251 (2002)
10. Diabat, A., Kannan, D., Kaliyan, M., Svetinovic, D.: An optimization model for product returns using genetic algorithms and artificial immune system. Resour. Conserv. Recycl. **74**, 156–169 (2013)
11. Dudek, G.: An artificial immune system for classification with local feature selection. IEEE Trans. Evol. Comput. **16**(6), 847–860 (2012)
12. Ernst, A.T., Krishnamoorthy, M.: Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem. Eur. J. Oper. Res. **104**(1), 100–112 (1998)
13. Filipović, V., Kratica, J., Tošić, D., Dugošija, D.: GA inspired heuristic for uncapacitated single allocation hub location problem. In: Mehnen, J., Köppen, M., Saad, A., Tiwari, A. (eds.) Applications of Soft Computing. Springer, Berlin Heidelberg (2009). https://doi.org/10.1007/978-3-540-89619-7_15
14. Grine, F.Z., Kamach, O., Sefiani, N.: A new efficient metaheuristic for solving the uncapacitated single allocation p-hub median problem. In: 2018 International Colloquium on Logistics and Supply Chain Management (LOGISTIQUA), pp. 69–74. IEEE (2018)
15. Ilić, A., Urošević, D., Brimberg, J., Mladenović, N.: A general variable neighborhood search for solving the uncapacitated single allocation p-hub median problem. Eur. J. Oper. Res. **206**(2), 289–300 (2010)

16. Kratica, J.: An electromagnetism-like metaheuristic for the uncapacitated multiple allocation p-hub median problem. Comput. Ind. Eng. **66**(4), 1015–1024 (2013)
17. Kratica, J., Stanimirović, Z., Tošić, D., Filipović, V.: Two genetic algorithms for solving the uncapacitated single allocation p-hub median problem. Eur. J. Oper. Res. **182**(1), 15–28 (2007)
18. Leung, C.S.K., Lau, H.Y.K.: A hybrid multi-objective ais-based algorithm applied to simulation-based optimization of material handling system. Appl. Soft Comput. **71**, 553–567 (2018)
19. Li, T., Song, R., He, S., Bi, M., Yin, W., Zhang, Y.: Multiperiod hierarchical location problem of transit hub in urban agglomeration area. Mathematical Problems in Engineering 2017 (2017), Article ID 7189060, 15 pages
20. Li, T., Song, R., He, S., et al.: Hierarchical model for regional integrated passenger hub layout. J. Beijing Univ. Technol. **40**(11), 1700–1706 (2014)
21. Martí, R., Corberán, Á., Peiró, J.: Scatter search for an uncapacitated p-hub median problem. Comput. Oper. Res. **58**, 53–66 (2015)
22. Meier, J., Clausen, U.: Solving classical and new single allocation hub location problems on euclidean data. Technical report, Optimization Online (2015)
23. Meier, J.F., Clausen, U., Rostami, B., Buchheim, C.: A compact linearisation of euclidean single allocation hub location problems. Electron. Notes Discrete Math. **52**, 37–44 (2016)
24. Milanović, M.: A new evolutionary based approach for solving the uncapacitated multiple allocation p-hub median problem. In: Gao, X.Z., Gaspar-Cunha, A., Köppen, M., Schaefer, G., Wang, J. (eds.) Soft Computing in Industrial Applications. Springer, Berlin Heidelberg (2010). https://doi.org/10.1007/978-3-642-11282-9_9
25. O'kelly, M.E.: A quadratic integer program for the location of interacting hub facilities. Eur. J. Oper. Res. **32**(3), 393–404 (1987)
26. Peiró, J., Corberán, Á., Martí, R.: Grasp for the uncapacitated r-allocation p-hub median problem. Comput. Oper. Res. **43**, 50–60 (2014)
27. Porselvi, S., Balaji, A., Jawahar, N.: Artificial immune system and particle swarm optimisation algorithms for an integrated production and distribution scheduling problem. Int. J. Logistics Syst. Manag. **30**(1), 31–68 (2018)
28. Rani, R.: Distributed query processing optimization in wireless sensor network using artificial immune system. In: Mishra, B.B., Dehuri, S., Panigrahi, B.K., Nayak, A.K., Mishra, B.S.P., Das, H. (eds.) Computational Intelligence in Sensor Networks, pp. 1–23. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-662-57277-1_1
29. Rostami, B., Meier, J., Buchheim, C., Clausen, U.: The uncapacitated single allocation p-hub median problem with stepwise cost function. Technical report, Optimization Online (2015)
30. Silva, D.G., Montalvão, J., Attux, R., Coradine, L.C.: An immune-inspired, information-theoretic framework for blind inversion of Wiener systems. Sig. Process. **113**, 18–31 (2015)
31. Sun, X., Dai, W., Zhang, Y., Wandelt, S.: Finding p-hub median locations: An empirical study on problems and solution techniques. J. Adv. Transp. 2017 (2017), Article ID 9387302, 23 pages
32. Wakui, T., Hashiguchi, M., Sawada, K., Yokoyama, R.: Two-stage design optimization based on artificial immune system and mixed-integer linear programming for energy supply networks. Energy **170**, 1228–1248 (2019)

33. Zhang, W., Yen, G.G., He, Z.: Constrained optimization via artificial immune system. IEEE Trans. Cybern. **44**(2), 185–198 (2014)
34. Zhang, Z., Yue, S., Liao, M., Long, F.: Danger theory based artificial immune system solving dynamic constrained single-objective optimization. Soft. Comput. **18**(1), 185–206 (2014)