Computer Science Journal of Moldova, vol.29, no.2(86), 2021

A Hybrid Method based on SA and VNS Algorithms for Solving DAP in DDS

Nasser Lotfi, Jamshid Tamouk

Abstract

Data allocation problem (DAP) is of great importance in distributed database systems (DDS). Minimizing the total cost of transactions and queries is the main objective of DAP which is mostly affected by the volume of transmitting data through the system. On the other hand, the volume of transmitting data depends on the fragment-to-site allocations method. DAP as a Np-hard problem has been widely solved by applying soft computing methods like evolutionary algorithms. In the continuation of our previously published research, this paper proposes a novel hybrid method based on Simulated Annealing Algorithm (SA) and Variable Neighborhood Search (VNS) mechanism for Solving DAP. To increase the performance, VNS mechanism is embedded into SA method in the proposed hybrid method. Technically speaking, in order to discover more promising parts of search space, the proposed method (VNSA) explores the search space via SA and fulfills more exploitation by applying neighborhood search mechanism. Moreover, due to the fact that both are a single solution-based method, they explore the search space faster than population-based methods. Performance of the proposed VNSA is experimentally evaluated using well-known benchmarks reported in state-of-the-art literature, and evaluation outcomes prove the robustness and fastness of the proposed hybrid method (VNSA). Furthermore, the results exhibit that VNSA outperforms its competitors and achieves better results in majority of test problems.

Keywords: Data Allocation Problem, Simulated Annealing, Variable Neighborhood Search.

1 Introduction

In DDS, minimizing the total cost of transactions and queries is of great importance which is mostly affected by amount of transmitting data through the system. Likewise, amount of transmitting data depends on fragment-to-site allocations method [1]. This NP-hard optimization problem in distributed systems is known as Data Allocation Problem (DAP). Data allocating methods in real world like sites, mail servers and search engines are significantly important because they deal with huge amount of data [2]. The detailed descriptions of DAP are given in Section 2. Due to it being an optimization problem, DAP has been widely solved by applying soft computing methods like evolutionary algorithms. The reason is that evolutionary algorithms are able to extract feasible and high quality solutions in acceptable computational time. Apart from a pretty wide literature can be found for solving DAP, two types of algorithms, namely static and dynamic algorithms, have been applied [3, 4, 32]. Taking the recent literature into account, researchers have been widely proposed hybrid methods for solving DAP [5]. The following part reviews some remarkable state-of-the-art methods for solving DAP.

Distributed database systems, fragmentation and allocation concepts were reviewed by Bhuyar et al. in [34]. Authors mentioned that fragmentation and allocation are two remarkable NP-hard challenges in DDS which can effectively affect performance of the system [34].

Wand et al. in [12] mapped DAP to well-known knapsack problem and solved it using Artificial Immune system method [13]. They also fulfilled comparison of the obtained results to some of state-of-the-art methods in order to measure the effectiveness of the applied method.

Biogeography-Based Optimization (BBO) method was applied by Singh et al. in [14] for minimizing total transmission and storage costs in database systems. Authors solved this fragment allocation problem and illustrated that their method is more effective than genetic algorithm [14].

Sen et al. [15] applied SA method for solving DAP and evaluated the method using standard benchmarks in CPLEX. Evaluation outcomes proved that SA is faster and more effective.

Genetic Algorithm (GA), Ant Colony Optimization (ACO) [18] and Tabu-search method [19] alongside a new crossover method were applied by Tosun in [17] for solving DAP. Also, effectiveness of the proposed method was evaluated using QAPLIB benchmarks.

Abdalla [30] proposed an innovative data allocation method for allocating fragments to sites based on communication costs. Authors also aimed to enhance transmission cost to improve distribution performance. They improved data fragmentation and allocation methods; they also carried out site clustering to produce minimum possible number of clusters. Performance of the proposed method was evaluated using TC objective function and results proved the efficiency of suggested method.

A hybrid method combining ACO and local search was proposed by Adl et al. [11] for solving DAP. Evaluation results indicated that the proposed method is flexible and successful in extracting near-optimal solutions. Ahmed et al. [21] suggested an evolutionary method for solving DAP. They also introduced a dependency graph for modeling fragments dependencies. To measure the efficiency, the suggested method was evaluated using standard benchmarks.

To solve DAP, Mahi et al. [2] applied Particle Swarm Optimization (PSO) for minimizing total transmission cost. Authors evaluated their results over 20 benchmarks; and outcomes proved that the suggested method, PSO-DAP, outperforms state-of-the-art methods in terms of quality and time. A clustering-based fragmentation method was developed by Sewisty et al. [25] which emphasizes on generating clusters. In the proposed method, disjoint fragments are generated using clusters. Evaluation results indicated that the introduced method is valid and effective. In order to minimize total transmission cost, Apers et al. [33] proposed a fragment allocation mechanism. Authors described details of DAP complexity and compared the obtained results to existent methods.

Lotfi proposed a new hybrid method (DEVNS) in [35] based on differential evolution (DE) and VNS algorithms for solving data allocation problem. The author enhanced DE algorithm by proposing better selection and crossover operators as well as embedding VNS into DE to increase its performance. The effectiveness of DEVNS was evaluated over well-known benchmarks and compared to nine existent methods reported in literature. The obtained results indicated that DEVNS outperforms all competitors in 13 out of 20 benchmarks. Even though DEVNS outperforms its competitors, it has some drawbacks. The main drawback and weakness of DEVNS is that it is slow and time consuming. DE is a population-based evolutionary algorithm, and working on a population of solutions is time consuming. Moreover, when it is combined with another algorithm like VNS, it becomes even slower. Hence, there is no comparison in terms of execution time in [35]. Another drawback is that according to the reported results, DEVNS does not work well on small size problems.

This paper proposes an innovative hybrid method, named as VNSA from this point on, for solving data allocation problem (DAP). The suggested method consists of SA algorithm [6, 7, 8] and VNS [9, 10].

All state-of-the art methods in [1] are population-based algorithms. Hence the aim of this study is to use combination of single-solution-based methods in order to achieve better results. On the other hand, since the single-solution-based methods are faster and time-efficient, the proposed hybrid method obtains better results in acceptable time. According to the proposed hybrid strategy, SA algorithm is mixed up with efficient neighborhood search mechanism to improve the exploration and exploitation performance. In the modified SA, instead of choosing the neighbors randomly, an efficient VNS is applied to fulfill more exploitation over the solution. Description of VNS mechanism and details of how to apply these operators are given in the following sections.

Performance of the proposed hybrid method (VNSA) is experimentally evaluated using the well-known benchmarks reported in state-of-the-art literature. Evaluation outcomes prove the robustness of the VNSA and exhibits that system outperforms its competitors and achieves better results. Moreover, the fact that they are single-solution-based methods, both SA and VNS explore the search space faster than the population-based methods. Hence, the proposed hybrid method has been faster compared to state-of-the-art methods.

The rest of the paper is formed as following: The detailed description of Data Allocation problem and the proposed hybrid method are given in Section 2. Section 3 demonstrates the algorithm parameters, experimental and comparison results. Finally, the conclusions and some future research works are given in Section 4.

2 The proposed hybrid method for solving DAP in DDS

A detailed description of Data Allocation Problem is presented in this section. Likewise, it describes the proposed innovative hybrid method comprising modified SA algorithm [6, 7, 8] and VNS technique [9, 10] for solving DAP [33, 34]. Motivation and novelty of this study is to combine SA and VNS methods to increase the capability of discovering more promising parts of search space in acceptable time. These two methods are not able to achieve better results than state-of-the-art methods individually, but hybrid version of these algorithms is robust and works efficiently.

Every site in DDS deals with a part of database [3, 11]. During the running time, many transactions with different frequencies are given to the sites. For this reason, a huge amount of data is transmitted between the sites. Minimizing the total completion time of transactions is the main goal of DAP which is affected by data transmission speed [11]. Parameter notations in DAP are described in Table 1 [11].

As it was mentioned above, minimization of total cost is the main goal in DAP. During the process, storage capacity and data transmission cost are considered as problem limitation and problem cost respectively [31]. Figure 1 demonstrates the transaction-fragment and site-transaction dependencies. Total cost is mostly affected by data transmitting through the system. Variable X_{ij} is defined as below [11].

Table 1 Notation Description [11]

Notation	Description
n	Number of Sites
m	Number of fragments
s_i	Site number i
$SiteCap_i$	Capacity of s_i
$UC_{n\times n}$	Cost of data transmission between all sites
uc_{i1i2}	Data transmission cost transmitted from site s_i to site s_j
f_j	Fragment Number j
$fragSize_j$	Size of f_j
l	Number of transactions
t_k	Transaction Number k
$FREQ_{n\times l}$	Transactions Frequency
$freq_{ik}$	Frequency of t_k on s_i
$TRFR_{l\times m}$	Direct transaction-Fragment dependency
$trft_{kj}$	Volume of data transmission from site of f_j to the site which executes transaction t_k.
$Q_{l\times m\times m}$	Indirect transaction-fragment dependencies
q_{kj1j2}	Amount of data transmission from site of f_{j1} to the site of f_{j2}.
Ψ	Allocation scheme
Ψ_j	Site of f_j assigned in scheme Ψ
$COST(\Psi)$	Data transmission cost in Ψ
$COST1(\Psi)$	Data transmission cost obtained from $TRFR_{l\times m}$
$COST2(\Psi)$	Data transmission cost obtained from $Q_{l\times m\times m}$
$STFR_{n\times m}$	Site-fragment dependencies
$stfr_{ij}$	Amount of data from f_j accessed by s_i in time unit
$PARTIALCOST1_{n\times m}$	Fragment-site allocation $COST1(\Psi)$
$partialcost1_{ij}$	Cost of f_j allocated to s_i obtained from direct dependencies
$QFR_{l\times m\times m}$	Indirect transaction-fragment dependencies considering frequencies
qfr_{kj1j2}	Amount of data transmission from site of f_{j1} to site of f_{j2} considering frequency of t_k
$FRDEP_{m\times m}$	Inter fragment dependencies
$frdep_{j1j2}$	Amount of data transmission from site of f_{j1} to site of f_{j2} considering indirect dependencies

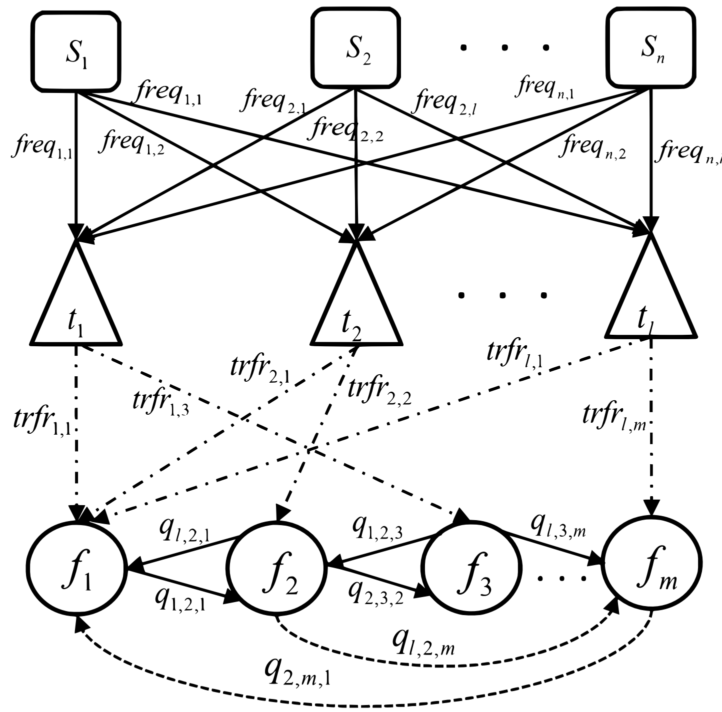

Figure 1. Transaction-fragment and site-transaction dependencies

$$X_{ij} = \begin{cases} 1, & \text{if } \Psi_j = s_i \text{ ;} \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where ψ_j is the site, where f_i is assigned. Hence, the storage capacity limitation is declared as Equation (2) [11]:

$$\sum_{j=1}^{m} fragSize_j \times X_{ij} \leq siteCap, i = 1, ..., n. \tag{2}$$

Therefore, total data transmissions cost is computed as below [11]:

$$COST(\psi) = \sum_{j=1}^{m} partialcost1_{\psi,j} = \sum_{j1=1}^{m} \sum_{j2=1}^{m} frdep_{j1,j2} \times uc_{\psi_{j1}\psi_{j2}}, \quad (3)$$

where $partialcost_{\psi 1}$ is the cost of storing f_i on site $s_{\psi j}$ which is calculated as Equation (4) [11]:

$$partialcost1_{\psi_j j} = \sum_{q=1}^{n} uc_{\psi,q} \times stfr_{qj}, \quad (4)$$

where $stfr_{qj}$ is amount of data from f_j accessed by s_q which is computed by Equation (5) [11]:

$$stfr_{qj} = \sum_{k=1}^{l} freq_{qk} \times trfr_{kj}. \quad (5)$$

Also, in the second part of the COST(ψ), the value of $frdep_{j1j2}$ is amount of data transferred from site f_{j1} to the site f_{j2} by taking the indirect dependencies into account. The direct transaction-fragment dependency (TRFR) is a matrix in which for each execution of t_k, the value of $trfr_{kj}$ represents amount of data transferred from site holding f_j to the site holding t_k. The dependency is called as direct dependency if for each execution of t_k there is some data transferred from site f_j to the site t_k. Value is calculated as Equation (6) [11]:

$$frdep_{j1j2} = \sum_{k=1}^{l} qfr_{kj1j2}, \quad (6)$$

where $qfr_{kj_1j_2}$ is volume of data transmitted from site holding f_{j1} to the site holding f_{j2}, which is calculated as Equation (7) [11]:

$$qfr_{kj1j2} = q_{kj1j2} \times \sum_{r=1}^{n} freq_{kr}. \quad (7)$$

The proposed hybrid method uses a modified SA to generate the neighbors effectively. In the modified SA, a new solution is generated by changing some sites randomly. Thereafter, more modification is carried out on new solution using VNS mechanism to discover more promising solutions around and replace them. Figure 2 demonstrates the flowchart for the proposed hybrid VNSA for solving DAP.

The flowchart in Figure 2 starts with parameter initialization in which the parameters are Neighborhood structure, Temperature, Cooling and Terminate. The neighborhood structure in VNS method is defined in a way that it makes somehow big modification over the solution. To do that, more sites are changed randomly to discover new solutions in far distance through the search space. This way, it would be possible to extract higher quality solutions through whole space. As well as temperature, cooling rate and terminate are initialized by 500, 0.2 and 0.1 respectively.

In the flowchart, there are some parameters – namely Y, K, Delta and P – in which Y is a new solution generated using neighborhood structure by changing site indexes, K is used as a counter for inner loop of VNS method, Delta is the difference between quality of current and new solution. The new solution is better if Delta holds positive value. Likewise, P is the acceptance probability of moving from current solution to a new generated solution.

In the next step, the current solution (allocation) is initialized by random. The solution (allocation) is represented as a one dimensional array with n columns, where n is the number of fragments in DDS. In this representation, the fragments and sites are shown by array indexes and array contents respectively. For instance, a sample allocation for 20 fragments and 4 sites is represented in Figure 3.

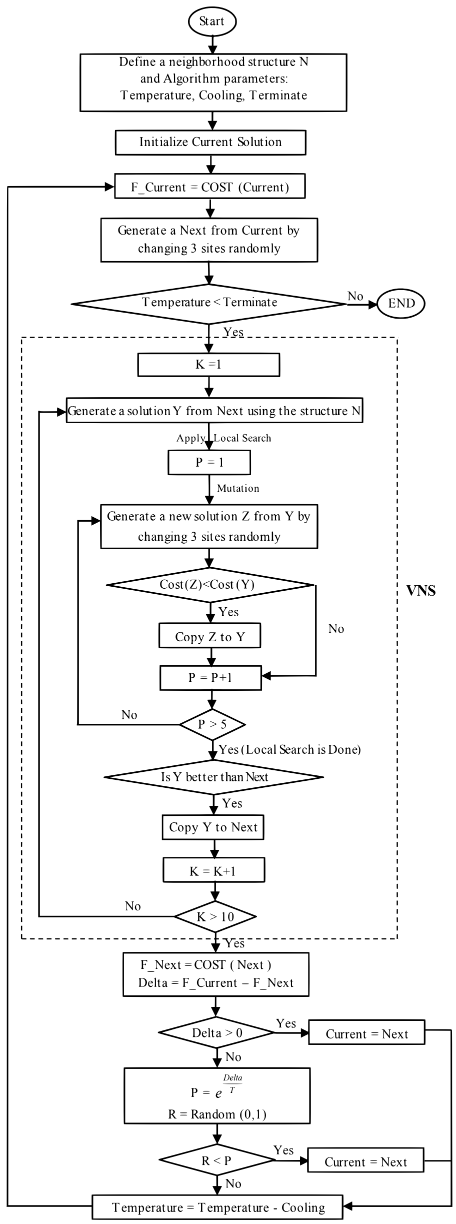

Figure 2. VNSA Flowchart

Index:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	s_1	s_2	s_3	s_1	s_2	s_1	s_4	s_2	s_2	s_1	s_4	s_4	s_3	s_1	s_1	s_4	s_2	s_3	s_3	s_4

Figure 3. A sample solution representation for 20 fragments and 4 sites

The first solution is initialized randomly using the algorithm shown in Figure 4.

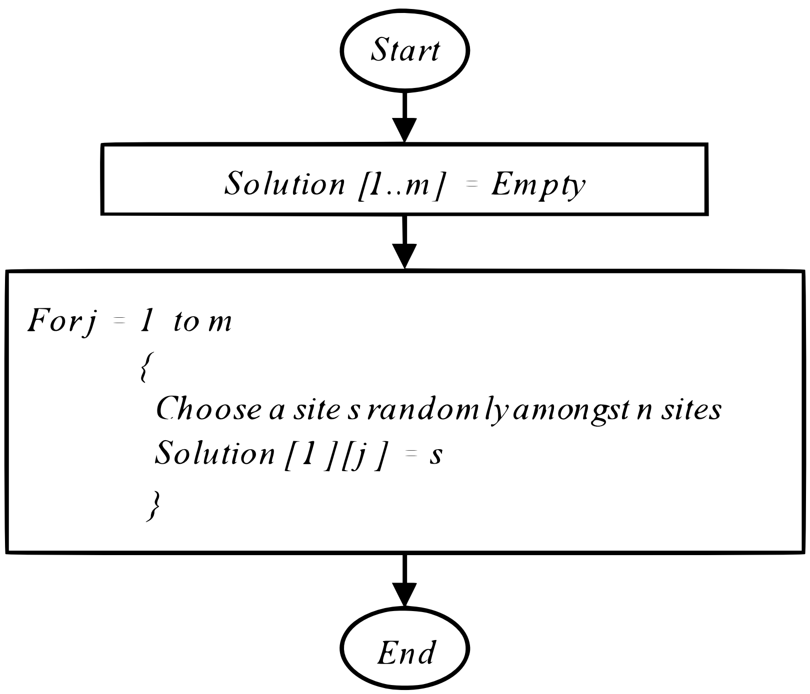

Figure 4. First solution initialization

Later on, system works in consecutive sessions until the temperature becomes less than termination value. The termination value is selected practically during the program running time. The termination value equal to 0.1 was adequate for the algorithm to execute fast and achieve good results. The total cost values are computed according to the descriptions and equations given in the beginning of Section 2. Algorithm for computing the total cost value for a solution is given in Figure 5.

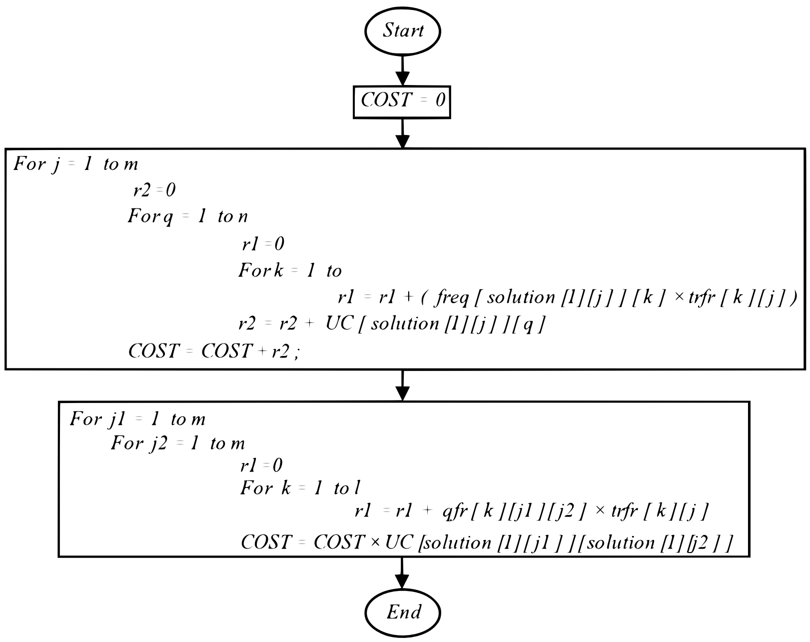

Figure 5. Cost value Calculation

Afterwards, a new solution named by 'Next' is generated by changing 3 sites on current solution randomly. Thereafter, VNS technique is applied over solution 'Next' to do more exploration and exploitation on search space. This way, solution 'Next' is adjusted and becomes more accurate. VNS algorithm is presented in Figure 2. In order to prevent time consuming by VNS, the inner loop is iterated 10 times. SA and VNS individually are single-solution-based algorithms and they are of fast kind. In case of combination, VNS adds more computational costs to the hybrid system. In order to have still fast hybrid system, a small size of VNS is applied. That's reason why the main loop in VNS is executed 10 times. VNS method starts with solution 'Next' and jumps to somewhat far neighbor y (Exploration). Then it starts to search locally around solution y to find a solution better than solution 'Next'. If it finds such solution, 'Next' is replaced by it, otherwise it jumps from solution 'Next' to another solution y and continues until either better solution is found or loop is terminated.

Once the VNS part is terminated, VNSA continues with the rest of SA. In this part, first of all the cost of solution 'Next' and corresponding Delta value are calculated. Then, if Delta value is positive, algorithm decides to move directly from current to Next. Otherwise it will move to Next based on a probability value P. This way if a randomly generated value R is less than P, it moves to Next otherwise algorithm will continue with current solution. Afterwards, cooling value is subtracted from temperature value and VNSA continues with the next session.

3 Evaluation and Experimental results

Evaluation results and performance of the proposed VNSA are presented in this section. To evaluate VNSA, benchmark set reported in [1, 2] is taken into account. This benchmark set has been generated using the rules mentioned in Section 2 and has been used by all state-of-the-art methods in literature [11]. All parameter values commonly used by state-of-the-art methods are shown in Table 2. Moreover, the unit cost is considered between zero and one. Also, Number of fragments and sites are considered as equal [1].

Table 2 Parameter values used in proposed VNSA

Parameter Description	Notation	Value
Fragment Size	C	10
Transmission costs between two sites	UCN	[0-1]
Number of Transactions	L	20
Probability of transaction requested at a site	RPT	0.7
Probability of fragment accessed by transaction	APF	0.4
Probability of a transaction needing data transmission between two sites	APFS	0.025
Temperature	T	200
Cooling	C	0.1
Terminate	Tr	0.1

The proposed VNSA was developed in Matlab® programming language environment and performed over a system with 2.00 GHz CPU and 2 GB of memory.

The following state-of-the-art methods are compared to VNSA in terms of total cost and time: ACO (Ant Colony Optimization) [1, 27], RTS (Robust Tabu Search) [1, 26], GA (Genetic Algorithm) [1, 28], HG-MTS (Hybrid Genetic Multi-start Tabu Search) [1], PSO-DAP (Particle Swarm optimization) [2] and DEVNS (Differential Evolution Variable Neighborhood Search) [35]. The total cost achieved by all methods over DAP instances is presented in Table 3.

Table 3 Cost comparison of methods for different DAP instance sizes (cost value is column × 10^6)

Size	HG-MTS	GA3	GA2	GA1	RTS	ACO	PSO-DAP	DEVNS	VNSA
5	0.04	0.04	0.04	0.04	0.04	0.04	**0.02**	0.03	0.03
10	0.31	0.31	0.31	0.32	0.31	0.31	**0.05**	0.06	0.06
15	0.98	0.98	0.98	0.99	0.98	0.98	**0.41**	0.52	0.48
20	2.61	2.64	2.64	2.63	2.61	2.61	**0.77**	1.47	1.31
25	5.19	5.24	5.26	5.25	5.19	5.19	3.74	**3.35**	**3.35**
30	10.27	10.41	10.42	10.39	10.27	10.27	3.19	**2.98**	**2.98**
35	16.39	16.66	16.61	16.64	16.39	16.39	9.04	**8.51**	8.76
40	25.92	26.21	26.33	26.28	25.9	25.91	19.24	20.71	**19.05**
45	37.27	37.82	37.8	37.73	37.26	37.28	27.04	**25.56**	26.20
50	53.88	54.69	54.63	54.76	53.89	53.93	34.43	29.38	**28.76**
55	71.21	72.13	72.40	72.72	71.19	71.30	51.38	**46.19**	46.23
60	90.20	91.56	91.49	91.76	**90.16**	90.35	97.78	90.20	90.20
65	**112.08**	113.84	113.75	113.59	112.13	112.31	125.01	113.45	112.10
70	146.15	148.18	148.8	148.48	146.19	146.41	138.69	131.72	**130.55**
75	177.65	180.63	180.75	180.04	177.7	177.90	171.47	168.34	**165.81**
80	**219.18**	222.96	222.80	223.10	219.26	219.40	260.86	234.26	223.04
85	261.99	266.19	266.15	267.04	261.88	262.24	260.63	260.33	**260.12**
90	315.86	320.58	320.93	320.88	315.86	316.11	287.09	**279.49**	283.04
95	369.91	375.29	375.85	375.49	369.92	370.14	365.06	355.04	**352.93**
100	427.98	434.45	436.15	436.19	428.28	428.40	481.58	424.08	**421.32**

Even though it can be noticed from Table 3 that HG-MTS, RTS, PSO-DAP and DEVNS are better that VNSA in 2, 1, 4 and 6 problems respectively, differences between results are very small. Likewise, VNSA is better than all methods in Table 3 for 9 problem instances. In order to check similarity of VNSA to other eight methods as well as to indicate the rank of VNSA among 10 methods, Friedman Aligned Rank test is performed. The test is carried out based on procedure described in [22, 23, 24, 30]. All ranks assigned to all problem-method pairs by Friedman Aligned Rank test are illustrated in Table 4.

Table 4 Friedman aligned ranks for all problem-method pairs.

Size	HG-MTS	GA3	GA2	GA1	RTS	ACO	PSO-DAP	DEVNS	VNSA
5	118	118	118	118	118	118	110	112	112
10	85	85	85	86	85	85	76	78	78
15	50	50	50	51	50	50	43	45	44
20	25	28	28	26	25	25	19	21	20
25	6	7	9	8	6	6	3	2	2
30	15	17	18	16	15	15	12	11	11
35	133	138	135	137	133	133	68	64	65
40	130	134	139	136	128	129	70	72	69
45	141	145	144	143	140	142	59	56	57
50	156	165	163	166	157	159	38	34	32
55	152	164	168	170	151	154	37	31	33
60	102	121	119	123	99	103	173	102	102
65	91	109	108	107	93	94	178	106	92
70	125	149	161	153	126	127	62	42	40
75	120	160	162	148	122	124	67	58	52
80	60	74	73	79	61	63	179	177	75
85	98	158	155	167	95	105	88	87	80
90	147	174	176	175	147	150	30	22	29
95	96	169	172	171	97	104	66	39	35
100	53	71	89	90	54	55	180	41	36

Table 5 shows the rank averages, FAR value and p value. To calculate FAR, the equation in [23] with statistical significance of χ^2 distribution and k-1 degrees of freedom is used, where k denotes number of methods. Likewise to indicate the significant difference between all methods, p value is calculated. According to Table 6, rank average for VNSA is smaller than others which indicates that VNSA is best performing algorithm. Also, DEVNS and PSO-DAP take the second and third places respectively. It can be seen that p value is very close to zero which shows that there is significant statistical difference between all methods in Table 5. Likewise, very small p value implies that VNSA is statistically different than other 8 methods.

Table 5 Average Friedman aligned ranks, FAR and p value

Method	Average Friedman aligned ranks
HG-MTS	95.15
GA3	111.8
GA2	113.6
GA1	113.5
RTS	95.1
ACO	97.05
PSO-DAP	77.9
DEVNS	60
VNSA	**53.2**
FAR	**56.3264**
P values	**0.0012**

Running times of all methods for 20 problem instances are given in Table 6.

Table 6 Running time comparison of methods for increasing DAP instance sizes.

Size	HG-MTS	GA3	GA2	GA1	RTS	ACO	PSO-DAP	DEVNS	VNSA
5	1.44	88.11	56.11	76.27	0.83	9.26	**0.74**	5.41	0.81
10	2.45	94.91	50.37	87.80	2.73	14.52	**1.35**	4.92	1.55
15	2.65	104.13	66.22	90.76	5.66	13.74	**2.17**	5.39	2.72
20	4.17	167.22	84.13	123.79	8.89	17.91	3.65	11.45	**3.12**
25	5.21	125.30	81.96	131.98	14.52	25.86	4.25	13.07	**3.91**
30	7.38	137.02	104.64	132.46	20.89	31.17	6.45	16.50	**5.45**
35	10.73	151.02	111.87	150.06	29.06	43.31	6.81	14.27	**6.23**
40	15.60	173.21	128.75	166.80	37.05	56.59	8.85	22.43	**7.56**
45	20.80	202.10	159.10	191.93	48.67	80.92	8.42	20.41	**6.89**
50	26.80	359.57	207.56	471.98	62.74	105.33	9.60	23.45	**8.58**
55	27.22	261.71	201.43	258.31	76.07	126.00	13.74	27.78	**12.15**
60	39.56	290.46	208.37	315.31	91.79	166.55	**16.09**	35.43	16.72
65	48.92	336.01	284.08	421.93	109.20	204.35	16.09	38.20	**15.56**
70	63.13	358.03	34.20	536.15	131.54	320.62	17.34	27.12	**17.05**
75	73.41	380.81	379.07	609.77	155.31	309.51	17.01	26.69	**15.31**
80	87.84	416.18	331.17	464.17	193.63	396.18	16.29	33.27	**14.39**
85	102.79	586.21	364.71	532.05	195.80	807.43	**18.31**	39.61	18.68
90	123.19	531.13	400.37	563.15	215.58	621.55	20.98	44.53	**18.76**
95	143.16	569.92	974.24	629.55	250.72	725.93	18.15	41.93	**16.38**
100	179.07	808.82	568.73	1236.30	278.63	1203.99	20.97	48.22	**20.73**

Table 6 indicates that the suggested VNSA outperforms all competitors in terms of speed. The reason is that VNSA is mixed of two single-solution-based algorithms, SA and VNS, which are much faster than population-based methods. Likewise, PSO-DAP is the second fastest method in which it is faster than VNSA in 5 out of 10 problem instances.

4 Conclusions and Future Works

This paper proposes an innovative hybrid method (VNSA) for solving data allocation problem in DDS. The method works based on a strategy to combine SA algorithm and VNS technique. To have an effective method, SA method is combined with effective neighborhood generation method which is called as VNS. The VNS technique is added to SA to provide more exploration and exploitation power and extract more accurate solutions. The obtained results demonstrate that the proposed hybrid method outperforms all state-of-the-art methods reported in literature. The proposed method was evaluated over different sizes of DAP in terms of cost and running time. According to the obtained results, VNSA took the first rank in 9 problems out of 20 problems in terms of cost value. Likewise, the results illustrate that VNSA is faster than all other methods under consideration. Also, the Friedman Aligned Rank test proved that significant difference between all methods exists, and VNSA is statistically different than all other methods. Future works are planned to add more details and parameters to DAP equations according to [25]. In new aspect of problem, Equation (3) will be enhanced by adding extra parameters like the number of sites involved in processing query and Communication costs between sites. Also variable X_{ij} will be inserted to Equation (3) to distinguish the existence of data fragment in the concerned site reached by the relevant query.

References

[1] U. Tosun, "Distributed database design using evolutionary algorithms," *Journal of Communications and Networks*, vol. 16, no. 4, pp. 430–435, 2014.

[2] M. Mahi, O. Baykan, and H. Kodaz, “A new approach based on particle swarm optimization algorithm for solving data allocation problem,” *Applied Soft Computing*, Elsevier, vol. 62, pp. 571–578, 2018. DOI: https://doi.org/10.1016/j.asoc.2017.11.019.

[3] I. Ahmad, K. Karlapalem, and Y. K. Kwok, “Evolutionary Algorithms for Allocating Data in Distributed Database Systems,” *Distributed and Parallel Databases*, vol. 11, pp. 5–32, 2012. DOI: https://doi.org/10.1023/A:1013324605452.

[4] A. Helal, “Dynamic Data Reallocation for Skew Management in Shared-Nothing Parallel Databases,” *Distributed and Parallel Databases*, vol. 5, pp. 271–288, 1997.

[5] J. Liu, S. Zhang, C. Wu, J. Liang, X. Wang, and K. L. Teo, “A hybrid approach to constrained global optimization,” *Applied Soft Computing*, Elsevier, vol. 47, no. C, pp. 281–294, 2016. DOI: 10.1016/j.asoc.2016.05.021.

[6] Y. Zhou, J. Wang, Z. Qiu, Z. Bi, and Y. Cai, “Differential evolution with guiding archive for global numerical optimization,” *Applied Soft Computing*, Elsevier, vol. 43, no. C, pp. 424–440, 2016. DOI: https://doi.org/10.1016/j.asoc.2016.02.011.

[7] K.V. Price, ”An Introduction to Differential Evolution,” in *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. London, UK: McGraw-Hill, 1999, pp. 79–108.

[8] R. Storn and K. Price, “Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 11, pp. 341-–359, 1997. DOI: https://doi.org/10.1023/A:1008202821328.

[9] P. Hansen and N. Mladenovic, “Variable neighborhood search: Principles and Applications,” *European Journal of Operational Researchs*, Elsevier, vol. 130, no. 3, pp. 449–467, 2001. DOI: https://doi.org/10.1016/S0377-2217(00)00100-4.

[10] L. Liberti and M. Drazic, "Variable Neighbourhood Search for the Global Optimization of Constrained NLPs," in *Proceeding of GO*, 2005, pp. 1–5. Corpus ID: 15444192.

[11] R. K. Adl and S. M. T. R. Rankoohi, "A new ant colony optimization based algorithm for data allocation problem in distributed databases," *Knowledge and Information Systems*, vol. 20, no. 3, pp. 349–373, 2009. DOI: 10.1007/s10115-008-0182-y.

[12] M. Wang, S. Feng, C. Ouyang, and Z. Li, "RFID tag oriented data allocation method using artificial immune network," in *27th Chinese Control and Decision Conference (2015 CCDC)*, IEEE, pp. 5218–5223, 2015.

[13] D. Dasgupta, S. Yua, and F. Nino, "Recent Advances in Artificial Immune Systems: Models and Applications," *Applied soft computing*, Elsevier, vol. 11, pp. 1574–1587, 2011.

[14] A. Singh, K. S. Kahlon, and R. S. Virk, "Replicated static allocation of fragments in distributed database design using biogeography-based optimization," in *Int. Conf. on Advances in Communication, Network, and Computing, CNC*, 2014, pp. 462–472.

[15] G. Sen, M. Krishnamoorthy, N. Rangaraj, and V. Narayanan, "Mathematical models and empirical analysis of a simulated annealing approach for two variants of the static data segment allocation problem," *Networks*, pp. 4–22, 2016.

[16] V. F. Yu, A. A. N. P. Redi, Y. A. Hidayata, and O. J. Wibowo, "A simulated annealing heuristic for the hybrid vehicle routing problem," *Applied Soft Computing*, Elsevier, pp. 119–132, 2017.

[17] U. Tosun, "A new recombination operator for the genetic algorithm solution of the quadratic assignment problem," *Procedia Computer Science*, vol. 32, pp. 29–36, 2014. DOI: https://doi.org/10.1016/j.procs.2014.05.394.

[18] R. Jovanovic, M. Tuba, and S. Vob, "An ant colony optimization algorithm for partitioning graphs with supply and demand," *Applied Soft Computing*, Elsevier, pp. 317–330, 2017.

[19] P. Cortes, J. Munuzuri, L. Onieva, and J.Fernandez, "A Tabu Search algorithm for dynamic routing in ATM cell-switching networks," *Applied Soft Computing*, Elsevier, pp. 449–459, 2011.

[20] H. I. Abdalla, "A new data re-allocation model for distributed database systems," *International Journal of Database Theory and Application*, vol. 5, no. 2, pp. 45–60, 2012. Corpus ID: 741957.

[21] I. Ahmad, K. Karlapalem, Y. K. Kwok, and S. K. So, "Evolutionary algorithms for allocating data in distributed database systems," *Distributed And Parallel Databases*, vol. 11, no. 1, pp 5–32, 2002. DOI: http://dx.doi.org/10.1023/A:1013324605452.

[22] U. Tosun, T. Dokeroglu, and A. Cosar, "Heuristic algorithms for fragment allocation in a distributed database system," in *Computer and Information Sciences III*, London: Springer, 2013, ch. 41, pp. 401–408. DOI: https://doi.org/10.1007/978-1-4471-4594-3_41.

[23] J. Derrac, S. Garcia, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evolutionary Computing*, vol. 1, no. 1, pp. 3–18, 2011.

[24] A. Acan and N. Lotfi, "A multiagent, dynamic rank-driven multideme architecture for real-valued multiobjective optimization," *Artificial Intelligence Reviews*, Springer Netherlands, vol. 48, no. 1, pp. 1–29, 2017. DOI: https://doi.org/10.1007/s10462-016-9493-7.

[25] A. Sewisy, A. Amer, and H. Abdalla, "A Novel Query-Driven Clustering-Based Technique for Vertical Fragmentation and Allocation in Distributed Database Systems," *International Journal on Semantic Web and Information Systems*, vol. 13, no. 2, pp. 27–54, 2017. DOI: 10.4018/IJSWIS.2017040103.

[26] E. Taillard, "Robust taboo search for the quadratic assignment problem," *Parallel Computing*, vol. 17, no. 4-5, pp. 443–455, 1991. DOI: https://doi.org/10.1016/S0167-8191(05)80147-4.

[27] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating Agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, Feb. 1996. DOI: 10.1109/3477.484436.

[28] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Boston, MA, United States: Addison-Wesley Longman Publishing Co., Inc., 1989, 372 p. ISBN:978-0-201-15767-3.

[29] D. Nashat and A. Amer, "A Comprehensive Taxonomy of Fragmentation and Allocation Techniques in Distributed Database Design," *ACM Computing Surveys*, pp. 1–25, 2018. Article No.: 12, DOI: https://doi.org/10.1145/3150223.

[30] H. Abdalla and A. M. Artoli, "Towards an Efficient Data Fragmentation, Allocation, and Clustering Approach in a Distributed Environment," *Information*, vol. 10, no. 3, 2019. Article no. 112. DOI: 10.3390/info10030112.

[31] A. Sinha and P. Jana, "A hybrid MapReduce-based k-means clustering using genetic algorithm for distributed datasets," *Journal of Supercomputing*, vol. 74, pp. 1562–1579, 2018.

[32] A. A. Amer, M. H. Mohamed, and K. Al Asri, "ASGOP: An Aggregated Similarity-Based Greedy-Oriented Approach for Relational DDBSs Design," *Heliyon*, vol 6, no. 1, 2020. pp. e03172. DOI: https://doi.org/10.1016/j.heliyon.2020.e03172.

[33] P. M. J. Apers, "Data Allocation in Distributed Database Systems," *ACM Transactions on Database Systems*, vol. 13, no. 3, pp. 263–304, 1988. DOI: https://doi.org/10.1145/44498.45063.

[34] P. R. Bhuyar and A. D. Gawande, "Distributed Database Fragmentation and Allocation," *Journal of Data Mining and Knowledge Discovery*, vol. 3, no. 1, pp. 58–64, 2012. ISSN: 2229-6662 & ISSN: 2229-6670.

[35] N. Lotfi, "Data allocation in distributed database systems: a novel hybrid method based on differential evolution and variable neighborhood search," *SN Applied Sciences*, vol. 1, Article no. 1724, 2019. DOI: https://doi.org/10.1007/s42452-019-1787-3.

Nasser Lotfi, Jamshid Tamouk

Received December 14, 2020
Accepted December 25, 2020

Nasser Lotfi
Faculty of Engineering, Cyprus Science University, Girne, N.
Cyprus via Mersin 10, Turkey
E–mail: `nasser.lotfi2020@gmail.com`

Jamshid Tamouk
Faculty of Engineering, Eastern Mediterranean University,
Famagusta, N. Cyprus via Mersin 10, Turkey
E–mail: `jamshid.tamouk20@gmail.com`