Research Article

Data allocation in distributed database systems: a novel hybrid method based on differential evolution and variable neighborhood search

Nasser Lotfi[1]

Received: 28 May 2019 / Accepted: 26 November 2019 / Published online: 29 November 2019

Abstract
Data allocation problem (DAP) in distributed database systems is a NP-hard optimization problem with significant importance in parallel processing environments. The solution of problem aims to minimize the total cost of transactions and settlement of queries in which he main cost regards to the data transmission through the distributed system. These costs are affected by the strategy how to allocate fragments to the sites. Researchers have been solving this challenging problem by applying soft computing methods especially evolutionary algorithms. This study proposes a novel hybrid method based on differential evolution (DE) algorithm and variable neighborhood search (VNS) mechanism for solving DAP problem. The suggested hybrid method (DEVNS) aims to increase the performance of DE algorithm by applying effective selection and crossover operators. Moreover, DEVNS goals to improve the solutions found so far using VNS technique. By applying VNS, more promising parts of search space can be extracted. Eventually, the introduced DEVNS explores the search space via DE and fulfills more exploitation by neighborhood search mechanism. Performance of proposed DEVNS is experimentally evaluated against nine state-of-the-art methods using well-known benchmarks reported in literature. Obtained results exhibits that proposed DEVNS takes the first position for 13 of 20 test problems. Likewise, Friedman aligned rank test is carried out to demonstrate that there is significant statistical difference between all methods.

Keywords Data allocation problem · Distributed database systems · Differential evolution · Variable neighborhood search

1 Introduction

Data allocation problem (DAP) in distributed database systems is a NP-hard optimization problem with great importance in distributed environments. In DAP problem the main objective is to assign set of fragments to set of sites in order to minimize the total cost of transactions. The main cost in the system comes from data transmission across the distributed system. Therefore, the fragment allocation affects and increases the overall cost [1]. In real world, it is obvious that huge amount of data is used by different types of sites e.g. mail servers and search engines [2]. Hence, the way of data allocation to the sites is of remarkable importance in reducing the total cost. Detailed description of DAP problem is presented in next sections.

The optimization problems are significant challenge in majority of engineering applications in which researchers have been extensively solving DAP as challenging problem by applying soft computing methods especially evolutionary algorithms due to their capability in extracting good solutions in an acceptable computational time. Two types of algorithms namely static and dynamic algorithms have been applied to solve DAP problem [3, 4]. The literature review and state-of-the-art in solving DAP problem is given in following sections.

✉ Nasser Lotfi, nasserlotfi@gau.edu.tr | [1]Faculty of Engineering, Girne American University, N. Cyprus via Mersin 10, Girne, Turkey.

Recently, the hybrid methods have been proposed for solving optimization problems because they increase the search efficiency by combining the abilities of different methods [5–7]. This study proposes an innovative hybrid method, named as DEVNS from this point on, for solving data allocation problem (DAP). The suggested method consists of differential evolution (DE) algorithm [8–10] with efficient operators and variable neighborhood search technique (VNS) [11, 12].

Differential evolution (DE) is a population-based algorithm for solving optimization problems. DE attempts to improve candidate solutions iteratively regarding to their fitness values. These kinds of algorithms are called as metaheuristics in which they are able to discover huge search spaces in acceptable computational time. Also, they don't need much information about the problem being solved. DE like all other metaheuristics does not guarantee to find optimal solution but it hopefully can find feasible and good solutions for optimization problems. DE works on population of solutions and generates new solutions by combining solutions selected from population. The new generated solutions are evaluated in terms of fitness to decide on adding them to the population. If new solutions have higher quality, they are added to the population, otherwise they are discarded. This way, all solutions in population moves over the search space to discover better positions than previous positions [8–10].

Variable neighborhood search (VNS) is also a single-solution based metaheuristic for solving optimization problems. VNS discovers neighbors of candidate solution in far distance over search space then carry out local search around them. In case of finding better solution in terms of fitness, VNS moves there and forgets the previous solution. VNS is used to solve discrete and continuous type of linear and non-line optimization problems [11, 12].

According to the proposed hybrid strategy, DE algorithm is modified with more efficient selection and crossover operators. Likewise, an efficient neighborhood search mechanism is applied to a solution instead of mutation operator in each iteration. In the modified DE, instead of applying fully random-based selection operator, the solutions are selected still randomly but based on their fitness using roulette wheel selection approach. This way, the better solutions in terms of fitness will have more chance to be selected. However, the roulette wheel selection (RWS) operator will give a small chance for worse solutions to be selected. In selection phase, for each solution in population 3 solutions are chosen using RWS. Later on a solution is generated using crossover operator applied on selected solutions. Thereafter VNS algorithm is applied on generated solution and if it becomes better than solution in population, replacement is done.

Variable neighborhood search (VNS) technique is applied over generated solution in each iterations to do more exploitation. This way, generated solution is adjusted and becomes more accurate. The flowchart for VNS search method is presented in following sections. In order to prevent time consuming VNS, the inner loop is iterated 10 times. Neighborhood structure N in VNS method is defined in a way that it makes somewhat big modification on the solution. For that, S sites are selected and modified randomly. The details of operators and VNS method are given in the following sections.

Performance of proposed DEVNS is experimentally evaluated against nine state-of-the-art methods using well-known benchmarks reported in literature. Obtained results exhibits that proposed DEVNS takes the first position for 13 of 20 test problems. Likewise, Friedman aligned rank test is carried out to demonstrate that there is significant statistical difference between all methods.

The rest of the paper is formed as following: The detailed description of Data Allocation problem is given in Sect. 2. Section 3 presents the state-of-the-art methods for solving the DAP problem. The proposed novel hybrid method (DEVNS) is expressed in Sect. 4. Section 5 demonstrates the algorithm parameters, statistical analysis and comparison results. Finally, the conclusions and some future research works are given in Sect. 6.

2 Data allocation problem in distributed database systems

This section describes the data allocation problem (DAP) in distributed database systems in details. In a distributed database system including a number of sites, each site is responsible to support a part of database [13]. In such a system, a big amount of data transmission between the sites is needed due to having many transactions with diverse frequencies submitted to the sites. In DAP problem, the objective is to minimize the completion time and cost of transactions which is affected mostly by required time for data transmission due to site-fragment and inter-fragment relations [13]. Table 1 represents the description of parameter notations used in DAP problem [13]. More description about parameters can be found in [13].

The transaction-fragment and site-transaction dependencies are indicated in Fig. 1.

As already mentioned, the objective in DAP problem is to minimize the total cost. As problem constraints, the storage capacity of sites and as problem cost, the data transmission cost is taken into account. Data transmission through distributed system puts much pressure upon the system and increases the cost. Variable X_{ij} is declared as follows [13]:

Table 1 Notation description [13]

Notation	Description
n	# of sites
m	# of fragments
s_i	Site #i
$SiteCap_i$	Site s_i capacity
$UC_{n\times n}$	Unit data transmission cost between two sites
uc_{i1i2}	Transmission cost of data item from site s_i to site s_j
f_j	Fragment #j
$fragSize_j$	Fragment f_j size
l	# of transactions
t_k	Transaction #k
$FREQ_{n\times l}$	Frequency of transactions
$freq_{ik}$	Frequency of t_k on s_i
$TRFR_{l\times m}$	Direct transaction-fragment dependency
$trft_{kj}$	Amount of data transmission from site of f_j to the site executing 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
$COST\ 1(\Psi)$	Data transmission cost obtained from $TRFR_{l\times m}$
$COST\ 2(\Psi)$	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 unit time
$PARTIALCOS\,T1_{n\times m}$	Fragment-site allocation $COST\ 1(\Psi)$
$partialco\,st1_{ij}$	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

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

where Ψ_j is the site which f_j is assigned. Then the storage capacity limitation is defined as [13]:

$$\sum_{j=1}^{m} fragSize_j \times X_{ij} \le siteCap_i \quad i = 1, \ldots, n \tag{2}$$

The total cost of data transmission is calculated as follows [13]:

$$COST\ (\Psi) = \sum_{j=1}^{m} partialco\,st1_{\Psi_j j} + \sum_{j1=1}^{m}\sum_{j2=1}^{m} frdep_{j1j2} \times uc_{\Psi_{j1}\Psi_{j2}} \tag{3}$$

where in the first part, $partialco\,st1_{\Psi_j j}$ is the cost of storing f_j on site s_{Ψ_j} which is calculated as Eq. 4 [13].

$$partialco\ st1_{\Psi_j j} = \sum_{q=1}^{n} uc_{\Psi_j q} \times stfr_{qj} \tag{4}$$

where $stfr_{qj}$ is amount of data from f_j accessed by s_q which is computed by Eq. 5 [13].

$$stfr_{qj} = \sum_{k=1}^{l} freq_{qk} \times trfr_{kj} \tag{5}$$

Also in the second part of $COST\ (\Psi)$, the value of $frdep_{j1j2}$ is amount of data from site of f_{j1} to the site of f_{j2} 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 $trfr_{kj}$ presents amount of data to be sent from site including f_j to the site including t_k. The dependency is called as direct dependency if for each execution of t_k there is some data sent from site of f_j to site of t_k.

The value of $frdep_{j1j2}$ is calculated as follows [13]:

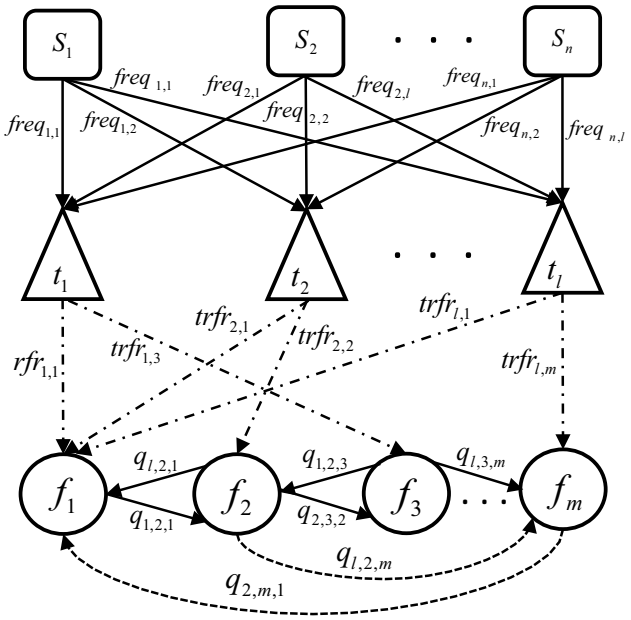


Fig. 1 Transaction-fragment and site-transaction dependencies

$$frdep_{j1j2} = \sum_{k=1}^{l} qfr_{kj1j2} \tag{6}$$

where qfr_{kj1j2} is amount of data sending from site of f_{j1} to the site of f_{j2} which is computed as Eq. 7 [13].

$$qfr_{kj1j2} = q_{kj1j2} \times \sum_{r=1}^{n} freq_{kr} \tag{7}$$

Section 4 explains our proposed hybrid method (DEVNS) for solving data allocation problem (DAP) using the cost function mentioned in Eq. 3.

3 State-of-the-art in methods for solving data allocation problem in distributed systems

Evolutionary methods including different metaheuristics have been extensively used to solve data allocation (DAP) problem. It can be found a pretty large literature in this field. This section reviews some significant state-of-the-art in methods for solving DAP problem.

In [14], the DAP problem was modeled as knapsack problem and then solved using Artificial Immune system method [15]. Authors compared their results to other existent methods to indicate the effects of memory capacity.

Singh et al. [16] proposed an innovative method for solving fragment allocation problem in distributed systems by applying biogeography-based optimization (BBO) approach. Authors used their suggested method to minimize the total transmission and storage costs and compared obtained results to genetic algorithm results. The evaluation results represent the robustness of the proposed method in [16] compared to genetic algorithm.

Sen et al. [17] solved DAP problem by applying simulated annealing (SA) algorithm [18] and evaluated their results using standard benchmarks in CPLEX. As evaluation result, they showed that SA algorithm works better in reasonable time.

Tosun [19], applied genetic algorithm (GA), ant colony optimization (ACO) [20] and Tabu-search method [21] to solve data allocation problem. The comparison results exhibited the robustness of suggested methods using QAPLIB benchmark library. Likewise, the author proposed a new Crossover method in his research.

A novel data allocation model in constraint distributed database systems was introduced in [22] by Abdalla. The suggested method allocates fragments to the sites according to communication and update costs for the fragments. Obtained results showed that the suggested method is more successful compared to state-of-the-art methods.

Adl et al. [13] suggested a method based on ACO and local search for solving DAP problem. The obtained results indicated that proposed method reaches to near-optimal solutions in acceptable computational time. The results also proved that proposed method is scalable and flexible.

Ahmad et al. [23] introduced a dependency graph for modeling the dependency between the fragments. The authors designed an evolutionary approach for data allocation in distributed database systems. The proposed method was evaluated over standard benchmarks to indicate the robustness of method.

A novel method based on particle swarm optimization (PSO) was proposed by Mahi et al. [2] for solving data allocation problem (DAP). The proposed method is appropriate for minimizing the total transmission cost. Suggested method, PSO-DAP, was evaluated over 20 different benchmarks and compared to existent methods in literature. Obtained results confirmed that PSO-DAP extracts solutions with higher quality rather than its competitors.

Bhuyar et al. [24] presented an introduction to distributed database systems in terms of fragmentation and allocation. According to [24], data fragmentation and allocation are two important issues in distributed database which are NP-hard problems. Authors mentioned that fragment allocation is critical issue for improving the performance of applications in distributed database systems.

In Sewisy et al. [25] developed a heuristic query-driven clustering-based vertical fragmentation technique which focuses on generates clusters. Later on they provide intended disjoint fragments using clusters. Experimental

results demonstrated that proposed method is effective and valid.

Abdalla et al. [26] aimed to enhance transmission cost to improve distribution performance. Authors improved data fragmentation and allocation methods as well as they carried out site clustering to produce minimum possible number of clusters. The performance of proposed method was evaluated using TC objective function and results proved the efficiency of suggested method.

In Apers et al. [27] proposed a method to minimize total transmission cost in which it carry out fragment allocation. Likewise, authors investigated the complexity of data allocation problem. Also they presented and compared methods to achieve optimal solutions.

4 The proposed hybrid method (DEVNS) for solving data allocation problem (DAP) in distributed database systems

This section describes the proposed innovative hybrid method (DEVNS) comprising modified differential evolution (DE) algorithm [8–10] and variable neighborhood search (VNS) technique [11, 12] for solving data allocation problem (DAP). The proposed DEVNS uses a modified DE in terms of selection, crossover and mutation operators. Figure 2 demonstrates the flowchart of proposed DEVNS for solving DAP problem.

The flowchart in Fig. 2 starts with initialization of population by random. In the proposed DEVNS, a solution (allocation) is represented as a one dimensional array with m columns, where m is the number of fragments in distributed database system. In this representation, 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 Fig. 3.

In the next step, population is initialized randomly using the algorithm shown in Fig. 4. Population is a two dimensional array in which the number of solutions and number of fragments are PopSize and m respectively. Population initialization algorithm in Fig. 4 fills this array with numbers between 1 and number of sites.

Later on, the system works in consecutive sessions until the termination criteria are met. In the first step of session, the system calculates the total cost values according to the descriptions and equations given in Sect. 2. The algorithm for computing the total cost value for a solution is given in Fig. 5.

Afterwards, the hybrid system (DEVNS) has an inner loop to do the exploration task using differential evolution algorithm. The loop repeats the following steps for each solution i in the population. In the first step, three different solutions are selected using roulette wheel selection by taking the total cost values into account. The algorithm is given in Fig. 2. Thereafter in three consecutive steps the crossover operator is carried out over randomly selected solutions to find out the final solution C. Crossover operator is carried out in a way that it generates feasible solutions. For this, the two-point crossover is applied to combine the sites allocated to fragments. Figure 6 indicates the crossover algorithm.

Afterwards, variable neighborhood search (VNS) technique is applied over solution C to do more exploration and exploitation. This way, solution C is adjusted and becomes more accurate. Figure 7 indicates the flowchart for VNS search method. In order to prevent time consuming VNS, the inner loop is iterated 10 times. The neighborhood structure in VNS method is defined in a way that it makes somehow big modification on the solution. To do that, more sites are changed randomly to discover new solutions in far distance over search space. This way, it would be possible to extract higher quality solutions through whole space.

Once the inner loop is terminated, DEVNS continues with the next session if the termination criteria are not met.

5 Evaluation and experimental results

This section demonstrates the evaluation results of applying proposed DEVNS over well-known benchmarks reported in state-of-the-art literature [1, 2]. The proposed method is tested using the same data set used by state-of-the-art methods. Benchmark data sets are generated using the rules and equations defined in Sect. 2 and according to experimental environment described in [13]. It should be noticed that unit cost is chosen in rage of [0, 1]. Table 2 presents the values of all parameters related to the DEVNS [2, 13]. DEVNS has been implemented in Matlab® programming language environment and executed on computer with CPU 2.00 GHz, memory 2 GB and 32-bit Windows 10 operating system. In the evaluation process, the number of generations is taken into account less than values reported in related literature [1, 2]

According to state-of-the-art literature [1, 2], the number of fragments and sites are assumed to be equal by all algorithms. The state-of-the-art literature to be compared to proposed DEVNS are PSO-DAP [2], SA [28], ACO [1], RTS [1], GA [1] and HG-MTS [1]. Table 3 represents the cost comparison of methods over different DAP instance sizes. Results of best performing algorithms are indicated in bold.

It can be seen over Table 3 that only PSO-DAP and HG-MTS methods are better than DEVNS in 2 and 5 of 20 problem instances respectively. The proposed DEVNS

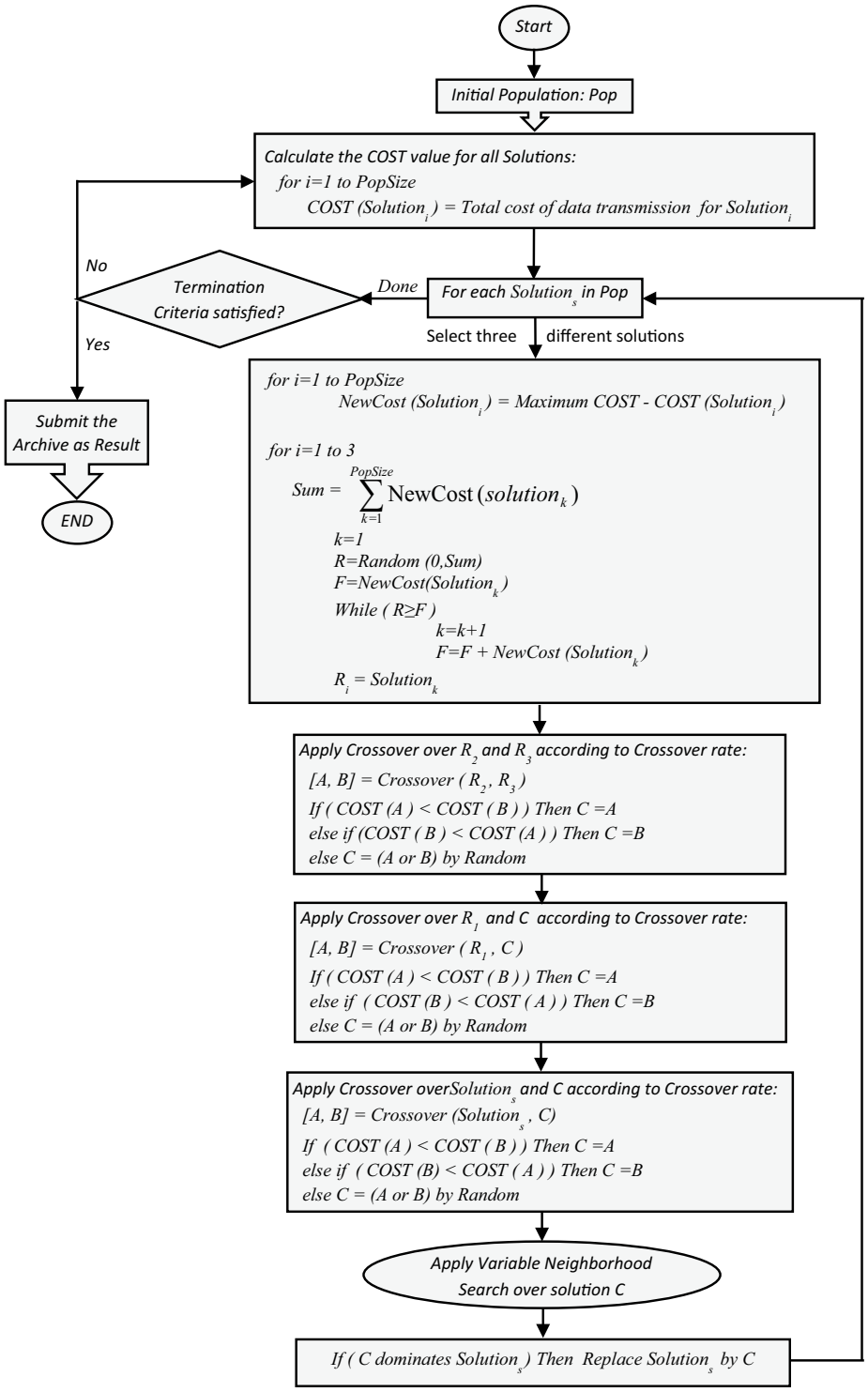


Fig. 2 Proposed hybrid method (DEVNS)

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

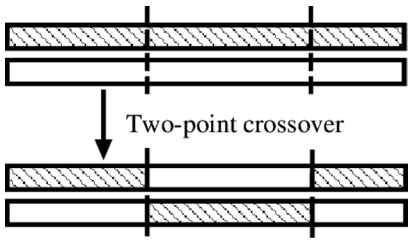
Fig. 3 A sample solution representation for 20 fragments and 4 sites

Fig. 4 Population initialization

Algorithm: Population-Initialization (*Solution [][], PopSize, m*) // m is number of fragments
1. For i = 1 to PopSize
   i. For j = 1 to m
      i. Choose a site s randomly amongst n sites // s= Random (1, s)
      ii. Solution [i][j] = s

Fig. 5 Total cost value calculation

Algorithm: Cost-Calculation (Solution [j][1..m]) // the cost of Solution j is calculated
1. COST = 0
2. r2=0
3. For q = 1 to m
   i. r1=0
   ii. For k = 1 to l
      i. r1 = r1 + (freq [solution [j]] [k] × trfr [k][j])
   iii. r2 = r2 + UC [solution [j]][q]
   ii. COST = COST + r2
2. For j1 = 1 to m
   i. r1=0
   ii. For k = 1 to l
      i. r1 = r1 + qfr [k][j1][j2] × trfr [k][j]
   iii. COST = COST × UC [solution [j][j1]][solution [j] [j2]



Algorithm: Schedule-Initialization (Parent1 [1..m] , Parent2 [1..m])
1. R=random(0,1)
2. if (R < CrossoverProbability)
   i. Cutpoint1 = RandomNumber (1,m)
   ii. Cutpoint2 = RandomNumber (1,m)
   iii. For i =1 to Cutpoint1
      i. Swap (Parent1 [i] and Parent2[i])
   iv. For i =Cutpoint2 to n
      i. Swap (Parent1 [i] and Parent2 [i])

Fig. 6 Two-point crossover

takes the first position for 13 of 20 test problems. Likewise, comparisons between PSO-DAP and DEVNS in the Table 3 indicates that mostly for the very small problem sizes, the PSO-DAP performs better, but the difference between obtained results is very small. When the problem size is growing, DEVNS performs much better than all competitors and PSO-DAP.

The last step of experimental evaluations is to perform the friedman aligned rank test. The rank test is performed over the all cost values obtained by all 9 methods presented in Table 3 and has two goals. The first goal is to check the similarity of DEVNS results to other 8 methods results. The second objective is to determine the order of DEVNS among all 9 methods. Friedman aligned rank test is performed according to the computational procedure explained in [29, 30]. Firedman aligned ranks for all 9 methods are presented in Table 4. This table indicates the ranks for all problem-method pairs.

The friedman aligned rank test statistic, FAR, is computed according to the formula described in [29] with statistical significance of χ^2 distribution and k-1 degrees of freedom where k is the number of methods. Meanwhile, the computed *p* value illustrates the significant differences between all methods under consideration. Eventually, the Table 5 indicates the average rank values, FAR and *p* value for all methods. It can be seen that average rank value for DEVNS is the smallest one which shows that DEVNS is the best performing method. For the same reason, the PSO-DAP is the second best performing method. Likewise, the *p* value is very close to zero that demonstrates that there is significant statistical difference between all methods. Meanwhile, a very small *p* value means that DEVNS is statistically different than other 8 methods.

Fig. 7 Variable neighborhood search method (VNS)

Algorithm: VNS (X [j][1..m])
1. Define a neighborhood structure N
2. For k = 1 to 10
   i. Generate a solution Y from X using the structure N
   ii. For p = 1 to 5
      i. Generate a new solution Z from Y by changing 3 sites randomly
      ii. If Z is better than Y
         i. copy Z to Y
   iii. If Y is better than X
      i. copy Y to X

Table 2 Parameter values used in proposed DEVNS

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
Population size	PopSize	200
Number of generations	GenSize	200
Crossover rate	PC	0.8
Mutation rate	PM	0.4

6 Conclusions and future works

This paper proposes an innovative hybrid method (DEVNS) for solving data allocation problem in distributed database systems. The method works based on a strategy to combine differential evolution (DE) algorithm and variable neighborhood search (VNS) technique. To have an effective method, DE method is modified with effective selection, crossover and mutation operators. Selection operator is carried out based on objective values instead of random selections to increase the chance of better solutions to be selected. Moreover, the variable neighborhood search technique is added to the algorithm instead of mutation operator to provide

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

Size	SA	HG-MTS	GA3	GA2	GA1	RTS	ACO	PSO-DAP	DEVNS
5	0.04	0.04	0.04	0.04	0.04	0.04	0.04	**0.02**	0.03
10	0.31	0.31	0.31	0.31	0.32	0.31	0.31	**0.05**	0.06
15	0.98	0.98	0.98	0.98	0.99	0.98	0.98	**0.41**	0.52
20	2.61	2.61	2.64	2.64	2.63	2.61	2.61	**0.77**	1.47
25	5.15	5.19	5.24	5.26	5.25	5.19	5.19	3.74	**3.35**
30	10.27	10.27	10.41	10.42	10.39	10.27	10.27	3.19	**2.98**
35	16.41	16.39	16.66	16.61	16.64	16.39	16.39	9.04	**8.51**
40	26.02	25.92	26.21	26.33	26.28	25.9	25.91	**19.24**	20.71
45	37.4	37.27	37.82	37.8	37.73	37.26	37.28	27.04	**25.56**
50	54.08	53.88	54.69	54.63	54.76	53.89	53.93	34.43	**29.38**
55	71.4	71.21	72.13	72.4	72.72	71.19	71.3	51.38	**46.19**
60	90.5	90.2	91.56	91.49	91.76	90.16	90.35	97.78	**90.2**
65	112.49	**112.08**	113.84	113.75	113.59	112.13	112.31	125.01	113.45
70	146.73	146.15	148.18	148.8	148.48	146.19	146.41	138.69	**131.72**
75	178.16	177.65	180.63	180.75	180.04	177.7	177.9	171.47	**168.34**
80	219.81	**219.18**	222.96	222.8	223.1	219.26	219.4	260.86	234.26
85	262.89	261.99	266.19	266.15	267.04	261.88	262.24	260.63	**260.33**
90	316.81	315.86	320.58	320.93	320.88	315.86	316.11	287.09	**279.49**
95	371.14	369.91	375.29	375.85	375.49	369.92	370.14	365.06	**355.04**
100	429.1	427.98	434.45	436.15	436.19	428.28	428.4	481.58	**424.08**

Table 4 Friedman aligned ranks for all problem-method pairs

Size	SA	HG-MTS	GA3	GA2	GA1	RTS	ACO	PSO-DAP	DEVNS
5	72	72	72	72	72	72	72	64	65
10	79	79	79	79	80	79	79	59	60
15	86	86	86	86	87	86	86	56	58
20	95	95	101	101	96	95	95	40	52
25	91	99	103	105	104	99	99	49	42
30	119	119	129	130	128	119	119	25	24
35	127	126	134	132	133	126	126	23	21
40	113	111	115	123	121	109	110	27	30
45	139	137	143	142	140	136	138	13	10
50	159	154	161	160	164	155	156	6	4
55	158	153	165	169	170	152	157	5	3
60	50	46	73	62	90	43	47	172	46
65	38	34	57	54	53	35	37	178	51
70	135	120	146	151	150	122	131	22	8
75	112	106	148	149	145	107	108	26	11
80	20	16	31	29	32	17	18	179	174
85	55	44	144	141	147	41	48	33	36
90	173	163	175	177	176	163	167	2	1
95	114	88	166	171	168	89	102	28	7
100	19	12	39	61	63	14	15	180	9

Table 5 Average Friedman aligned ranks, FAR and *p* value

Method	Average Friedman aligned ranks
SA	97.7
HG-MTS	93
GA3	113.35
GA2	114.7
GA1	115.95
RTS	92.95
ACO	95.5
PSO-DAP	59.35
DEVNS	35.6
FAR	69.3264
p values	0.0016

more exploration and exploitation power and extract more accurate solutions. Obtained results demonstrate that proposed hybrid method outperforms all state-of-the-art methods reported in literature. Future works are planned to use the proposed method with better strategies over different problems.

Compliance with ethical standards

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Tosun U (2014) Distributed database design using evolutionary algorithms. J Commun Netw 16:430–435
2. Mahi M, Baykan OK, Kodaz H (2018) A new approach based on particle swarm optimization algorithm for solving data allocation problem. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2017.11.019
3. Brunstrom A, Leutenegger S T, Simha R (1995) Experimental evaluation of dynamic data allocation strategies in a distributed database with changing workloads. In: Proceedings of the fourth international conference on information and knowledge management. ACM, pp 395–402
4. Gu X, Lin WJ, Veeravalli B (2006) Practically realizable efficient data allocation and replication strategies for distributed databases with buffer constraints. IEEE Trans Parallel Distrib 17:1001–1013
5. Liu J, Zhang S, Wu C, Liang J, Wang X, Teo KL (2019) A hybrid approach to constrained global optimization. Appl Soft Comput 47:281–294
6. Lotfi N, Acan A (2015) Learning-based multi-agent system for solving combinatorial optimization problems: a new architecture. In: 10th international conference on hybrid artificial intelligent systems (HAIS), pp 319–332
7. Lotfi N (2019) Ensemble of multi-objective metaheuristics for multiprocessor scheduling in heterogeneous distributed systems: a novel success-proportionate learning-based system. SN Appl Sci. https://doi.org/10.1007/s42452-019-1477-1
8. Zhou Y, Wang J, Zhou Y, Qiu Z, Bi Z, Cai Y (2016) Differential evolution with guiding archive for global numerical optimization. Appl Soft Comput 43:424–440
9. Dorigo D, Prize K, Glover F (1999) An introduction to differential evolution: new ideas in optimization. McGraw-Hill, Maidenherd

10. Storn R, Price K (1997) Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. J GlobOptim 11:341–359
11. Hansen P, Mladenovic N (2001) Variable neighborhood search: principles and applications. Eur J Oper Res 130:449–467
12. Liberti L, Drazic M (2005) Variable neighbourhood search for the global optimization of constrained NLPs. In: Proceeding of GO, pp 1–5
13. Adl RK, Rankoohi SMTR (2009) A new ant colony optimization based algorithm for data allocation problem in distributed databases. Knowl Inf Syst 20:349–373
14. Wang M, Feng S, Ouyang C, Li Z (2015) RFID tag oriented data allocation method using artificial immune network. In: 27th Chinese control and decision conference. IEEE, pp 5218–5223
15. Dasgupta D, Yua S, Nino F (2011) Recent advances in artificial immune systems: models and applications. Appl Soft Comput 11:1574–1587
16. Singh A, Kahlon K S, Virk R S (2014) Replicated static allocation of fragments in distributed database design using biogeography-based optimization. In: Proceeding of international conference on advances in communication, network, and computing, CNC, pp 462–472
17. Sen G, Krishnamoorthy M, Rangaraj N, Narayanan V (2016) Mathematical models and empirical analysis of a simulated annealing approach for two variants of the static data segment allocation problem. Networks 68:4–22
18. Yu VF, Redi AANP, Hidayat YA, Wibowo OJ (2017) A simulated annealing heuristic for the hybrid vehicle routing problem. Appl Soft Comput 53:119–132
19. Tosun U (2014) A new recombination operator for the genetic algorithm solution of the quadratic assignment problem. Procedia Comput Sci 32:29–36
20. Jovanovic R, Tuba M, Vob S (2017) An ant colony optimization algorithm for partitioning graphs with supply and demand. Appl Soft Comput 41:317–330
21. Cortes P, Munuzuri J, Onieva L, Fernandez J (2011) A Tabu search algorithm for dynamic routing in ATM cell-switching networks. Appl Soft Comput 1:449–459
22. Abdalla HI (2012) A new data re-allocation model for distributed database systems. Int J Database Theory Appl 5:45–60
23. Ahmad I, Karlapalem K, Kwok YK, So SK (2002) Evolutionary algorithms for allocating data in distributed database systems. Distrib Parallel Databases 11:5–32
24. Bhuyar PR, Gawande AD (2012) Distributed database fragmentation and allocation. J Data Min Knowl Discov 3(1):58–64
25. Sewisy A, Amer A, Abdalla H (2017) A novel query-driven clustering-based technique for vertical fragmentation and allocation in distributed database systems. Int J Semant Web Inf Syst 13(2):27–54
26. Abdalla H, Artoli AM (2019) Towards an efficient data fragmentation, allocation, and clustering approach in a distributed environment. Information 10:112
27. Apers PMJ (1966) Data allocation in distributed database systems. ACM Trans Database Syst 13(3):263–304
28. Tosun U, Dokeroglu T, Cosar A (2013) Heuristic algorithms for fragment allocation in a distributed database system. In: Gelenbe E, Ricardo L (eds) 27th international symposiumon computer and information sciences (ISCIS). Computer and information sciences III. Springer, pp 401–408
29. Derrac J, Garcia S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evolut Comput 1:3–18
30. Lotfi N, Acan A (2017) A multi-agent dynamic rank-driven multi-deme architecture for real-valued multi-objective optimization. Artif Intell Rev 48:1–29