

The Algorithm for Sequential Analysis of Variants for Distribution of Virtual Machines in Data Center

Oleksandr Rolik, Maksym Bodaniuk, Valerii Kolesnik
Department of Automation and Control in Technical
Systems
National Technical University of Ukraine “Igor Sikorsky
Kyiv Polytechnic Institute”
Kyiv, Ukraine
e-mail: o.rolik@kpi.ua, max.bodaniuk@gmail.com,
kolesnik.valerii@gmail.com

Volodymyr Samoty
Department of Automation and Information Technologies
Faculty of Electrical and Computer Engineering,
Cracow University of Technology, Cracow, Poland
Lviv State University of Life Safety, Lviv, Ukraine
e-mail: vsamoty@pk.edu.pl

Abstract—this work proposes an algorithm of sequential analysis of variants (SAV) to solve the distributional problem of allocation of virtual machines to physical servers in a data center. The set of tests and rules of the SAV algorithm is defined. The experimental results for problems of different dimensions are given. The comparison of the proposed algorithm with heuristic and genetic algorithms is accomplished. The time of finding solution required by the SAV algorithm depending on the dimension of the problem is evaluated. The recommendations for using the SAV algorithm are given. For tasks requiring high precision distribution it is better to use the SAV algorithm as it finds the optimal solution, whereas heuristic and evolutionary algorithms can quickly get an effective solution. The speed of the heuristic and evolutionary algorithms is not significantly dependent on the problem’s size, but the quality of their solutions is worse than equivalent solution received with the SAV algorithm.

Keywords—VM allocation; data center; resource allocation; sequential analysis of variance, virtual machine distribution

I. INTRODUCTION

Virtualization of computational resources remains an essential attribute of modern data centers (DC). Maintaining of IT infrastructure in a DC requires significant financial expenses, which makes a problem of efficient resource allocation relevant. A maximal density of virtual machines (VM) allocation among physical servers (PS) allows operational expenses to be reduced and more users to be served with DC’s server farm. That is why current paper is dedicated to solve the problem of optimal VMs distribution among physical servers, which is a challenging task.

II. ANALYSIS OF LATTER WORKS

Nowadays, increasing interest can be seen in the area of algorithms, which work according to the most important natural law, namely evolution. One of the well-known algorithms in this area are genetic algorithms, evolutionary strategies and programming. These algorithms simulate the principle of evolution (survival of the fittest) [1].

Such algorithms maintain a population of potential solutions (individuals) through repeated application of some “evolutionary” operators. They yield individuals with successfully improved fitness, and converge, hopefully, to the fittest individuals representing optimal solutions [2].

In [3] the authors have considered the problem of managing the data center by a cloud service provider. The cloud service provider makes available a variety of services for the customers. Some services may be provided to other users by implementing multi-tenant data centers which represent a very complicated networking environment. Between the cloud service providers there are agreements on providing the services using different data centers. There are also the corresponding access policies to ensure the possibility of providing resources on demand.

The cloud service provider issues the following services: virtual machines, middle-boxes, services, service workflows. A separate virtual machine is allocated to provide a service.

A problem of VMs distribution among physical servers can be solved with classical genetic algorithm (CGA) [4-6]. Papers [4-6] show that usage of CGA can improve quality of problem solving and can control course of a solution on each iteration of algorithm. Heuristic algorithms can also be used for allocating VMs. Also, it should be mentioned that there are a few objective functions developed for VM distribution problem, including the energy consumption minimization [7], minimization of PS amount [8], network traffic minimization [9], availability maximization [10], resource utilization maximization [7] and other.

Additionally, those papers contain proof of relation between efficiency of heuristic algorithms and algorithm structure and parameters of distributed VMs. However, those publications do not propose an algorithm, which obtains optimal solution at VMs allocation problem.

III. GOAL OF THE RESEARCH

The goal of current research is to develop and define feasibility of application of proposed algorithm for sequential analysis of variants (SAV) which is used to solve problem of VMs distribution on physical servers of a DC.

IV. MATHEMATICAL MODEL

A data center contains an ordered set of physical servers $N = \{N_1, \dots, N_n\}$, where n is the number of servers; an ordered set of virtual machines $K = \{K_1, \dots, K_m\}$, where m is the number of VMs, which are subjected to distribution.

Each server N_i , $i = \overline{1, n}$, has the following parameters: Ω_i stands for processor performance or CPU load; Γ_i stands for RAM capacity of the server N_i .

Each VM K_j , $j = \overline{1, m}$, requests ω_j of CPU time and γ_j of RAM resource;

Matrix $R = \|r_{ji}\|$ with dimensions $m \times n$ defines a distribution of VMs among servers, moreover

$$r_{ji} = \begin{cases} 1, & \text{if VM } K_j \text{ is located on server } N_i, \\ 0, & \text{otherwise.} \end{cases}$$

And matrix R is a solution of the problem and it defines a distribution of set of VMs K over set of physical servers N .

V. PROBLEM STATEMENT

In case where all servers from set N have identical technical characteristics, it can be considered that $\Omega_i = 1$ and $\Gamma_i = 1$, $i = \overline{1, n}$, i.e.

$$\{\Omega_i, \Gamma_i\} |_{N_i} = \{1, 1\}, \quad i = \overline{1, n}. \quad (1)$$

We can see the transition from resource measurement on servers in absolute units where RAM is measured in megabytes (MB) and CPU frequency in megahertz (MHz), to relative units. Then, needs in resources of VMs from set K are defined as a normalized part of resources capacities of a server.

In the paper the task of clustering is not considered, that is why requirements to resources of each VM from set K cannot exceed capacities of a server

$$\omega_j \leq 1 \text{ and } \gamma_j \leq 1, \quad i = \overline{1, n}. \quad (2)$$

For solving the problem of VMs distribution, all servers from set N must satisfy the following resource constraint:

$$\sum_{j=1}^m r_{ji} \omega_j \leq 1 \text{ and } \sum_{j=1}^m r_{ji} \gamma_j \leq 1, \quad i = \overline{1, n}. \quad (3)$$

After that, next vector will be introduced $\bar{y} = \langle y_i \rangle$, $i = \overline{1, n}$, where

$$y_i = \begin{cases} 1, & \text{if server } N_i \text{ contains at least one VM,} \\ 0, & \text{otherwise.} \end{cases}$$

Then, optimum criteria for solving the problem of VMs distribution on physical servers will be:

$$\min \sum_{i=1}^n y_i. \quad (4)$$

In other words, virtual machines should allocate physical server in a way that the minimal amount of physical servers was involved.

Such approach allows to minimize total costs S for maintenance and power supply of DC servers. Objective function can be represented thus:

$$S = s \sum_{i=1}^n y_i, \quad (5)$$

where s is the cost of maintenance and power supply for one server.

Then problem statement can be represented with next interpretation: physical servers in DC should allocate resources for VM in a way that allows minimum value in expression (5) to be achieved.

VI. ALGORITHM OF SEQUENTIAL ANALYSIS OF VARIANTS FOR DISTRIBUTION OF VIRTUAL MACHINES

The algorithm of SAV relates to combinatorial methods of problems solving. The main idea of combinatorial methods is in the use of finiteness of a feasible set and substitution of its complete enumeration with shortened one (directional enumeration). General scheme for sequential analysis of variants is developed by Myhalevich V.S. [11] based on ideas of theory of sequential solutions and dynamic programming. Methods of SAV are based on systematic solutions construction, or in other words: incremental specification of component values of the solution, and elimination of solutions which cannot be considered as a part of the optimal one.

Let us consider a minimization problem:

$$\min f(x_1, \dots, x_m) = \min s \sum_{i=1}^n y_i. \quad (6)$$

With constraints in the following representation:

$$\left(\sum_{j=1}^s (r_{ji} \omega_j) y_1, \dots, \sum_{j=1}^s (r_{ji} \omega_j) y_n \right) \leq 1, \quad j = \overline{1, m}, \quad i = \overline{1, n}, \quad (7)$$

$$\left(\sum_{j=1}^s (r_j \gamma_j) y_1, \dots, \sum_{j=1}^s (r_j \gamma_j) y_n \right) \leq 1, \quad j = \overline{1, m}, \quad i = \overline{1, n}, \quad (8)$$

$$x_j \in Q, \quad j = \overline{1, m}, \quad (9)$$

where $Q = \{q_{1,j}, \dots, q_{\alpha_m, j}\}$ are given final sets [12]. Each variable $x = (j; i)$ for all $j = \overline{1, m}$ expresses a location of j -th virtual machine on i -th server.

Vector $\bar{x}_{(m)} = (x_1, \dots, x_m)$ is a solution if its components $x_j \in Q, \quad j = \overline{1, m}$. Each element of set $q_{\alpha_m, i}$ represents one of possible components of the vector $\bar{x}_{(m)}$.

A set of all solutions is defined as Z . A solution is acceptable if it satisfies inequalities (7) and (8). A set of all acceptable solutions will be defined with Z_f . Vector $\bar{x}_{(p)} = (x_1, \dots, x_p)$, $p < m$ is called partial solution [11] in case if $\bar{x}_{(p)} \in Q$. If under these conditions vector $\bar{x}_{(p)}$ can be constructed to the acceptable solution $(x_1, \dots, x_p, x_{p+1}, \dots, x_m)$, $\bar{x}_{(m)} \in Z_f$, then this vector is named an acceptable partial solution.

A set of all solutions represents a tree where path from root node to top nodes corresponds to partial solutions and path from the node x_0 to the node x_m corresponds to complete solutions (see Fig. 1).

For the problem when all servers can provide the same amount of resources it is advisable to except mirrored solutions from consideration. For example if there are two identical servers then the number of possible solutions of VMs allocation can be reduced twice because transfer of all VMs from the first server to the second server will not influence the amount of used resources. For servers with identical configuration, the relation between their location and location of others is important, but not an absolute location (see Fig. 2). With increasing number of servers with similar configuration the number of mirrored combination significantly grows from $(m!-1)$ for the case of VMs allocation on one server, to $(m \times n - 1)!$ for the case of VMs allocation on a different servers.

For algorithm development it is necessary to define a set of rules for selecting solutions which will be improving each on each step. Rules are set in a way of eliminating tests, which define strategy of development of partial solutions. The

definition of elimination tests set $\xi = \{\xi_i\}$ is necessary for selection of partial solutions, which cannot be improved either up to acceptable or to optimal solutions [12].

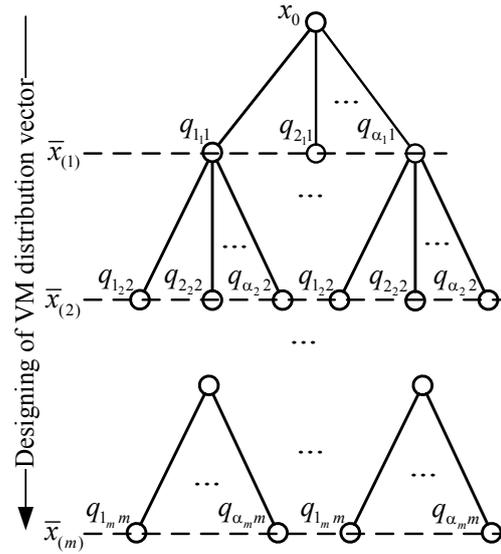


Fig. 1. Tree of solutions for algorithm SAV

Let h be some set of partial solutions. Then in ordered set $\sigma(h) = \{\xi_0(h), \xi_1(h), \dots, \xi_l(h)\}$ the variable $\xi_i(h)$ denotes a set of partial solutions, which are excepted with test ξ_i .

Let us next introduce a set of elimination tests:

ξ_0 — analysis of solutions admissibility;

ξ_1 — comparison of acceptable solutions by a value of objective function;

ξ_2 — evaluation of optimal value γ_f of objective function on each of the sets (calculation of lower bound).

The test ξ_0 allows to exclude unacceptable solutions from consideration. The set of unacceptable solutions contains all solutions for which an amount of allocated VMs on server exceeds the server's capacity. The test ξ_1 is designed to search and find optimal distribution of VMs and exclude only not optimal but acceptable solutions from consideration.

The test ξ_2 and the use of rule for excluding of mirrored solutions allow to avoid a complete enumeration of acceptable solutions.

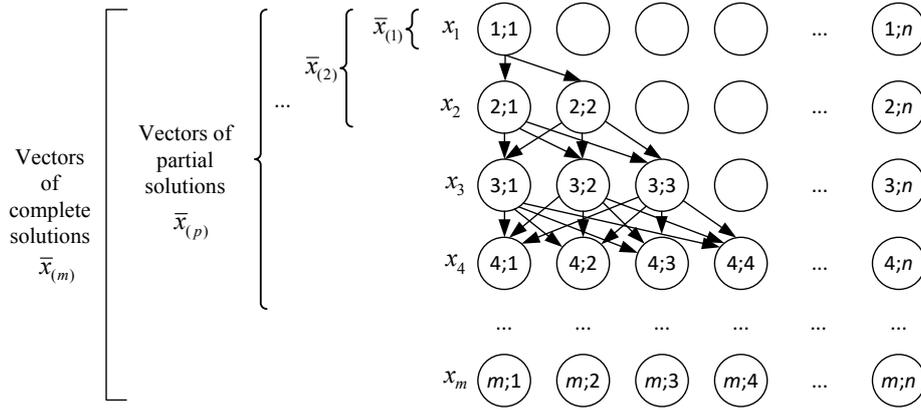


Fig. 2. Allocation of VMs on servers with the use of the SAV algorithm

Definition of elimination tests ξ_0 , ξ_1 , ξ_2 is represented below:

$$\xi_0(h) = \{\bar{x}_{(p)} \mid \bar{x}_{(p)} \in h, \sum_{j=1}^p x_{ji} \omega_j > 1, \sum_{j=1}^p x_{ji} \gamma_j > 1, i = \overline{1, n}\}, (10)$$

$$\xi_1(h) = \{\bar{x}_{(m)} \mid \bar{x}_{(m)} \in Z_f, f(x) \leq f(\bar{x}_{(m)})\}, (11)$$

$$\xi_2(h) = \{\bar{x}_{(p)} \mid \bar{x}_{(p)} \in h, \gamma_f(\bar{x}_{(p)}) > f^*\}, (12)$$

where f^* is the upper bound for problem minimum (6). The test ξ_2 calculates lower bound for optimal value of objective function $f(\bar{x}_{(p)}^0)$ on each of sets $H(\bar{x}_{(p)}^0)$ where elements are solutions, which extend partial solution $\bar{x}_{(p)}^0$.

The order of application of eliminating tests and rules of selection of partial solutions $U = (u_1, \dots, u_q)$ has impact on results of SAV. In SAV algorithm for distribution problem the set of rules U is defined by the following rules: u_1 , u_2 , u_3 .

The rule u_1 is defined as follows: on each step of algorithm, all partial solutions are developed. These solutions are defined on previous step beside the solutions, which are mirrors of previously considered solutions.

The rule u_2 describes next condition: on each step of the algorithm from all previously received partial solutions, only those are selected, which allow to achieve a minimum of objective function.

The rule u_3 is defined as follows: selection of candidate on each step of the scheme is performed from set $h_d = h_k - h_{k-1}$ of partial solutions, which are obtained on previous step and only in case when h_d does not contain a partial solution, it is necessary to look up whole list of solutions.

The set of rules $U = (u_1, u_2, u_3)$ and tests $\xi = \{\xi_0, \xi_1, \xi_2\}$ define a strategy of solutions enumeration. An order of rules and eliminating tests application corresponds to indexes. In the given case, rules and tests are ordered by ascending complexity.

VII. RESULTS OF EXPERIMENTAL RESEARCHES

A. Specifics of experiments

Requests for allocation of the set of VMs in DC are incoming sequentially. Amount of requested resources for each VM is chosen randomly. Dependency of required RAM and CPU is distributed uniformly.

During experimental researches a given amount of sets of VMs was generated. Needs in computational resources such as RAM or CPU for VMs are also generated randomly within bounds of 0,05...0,6 from server's capacity.

B. Results of experiments

Efficiency of SAV algorithm performance was evaluated by the parameter of time required for searching for optimal solution, depending on problem complexity m . Results are shown on Figure 3. For comparison, figure depicts time of manageable genetic algorithm (MGA) and heuristic algorithms "T" and "M", proposed in [4].

Figure 3 shows that SAV shows acceptable results of allocation time on problems with low complexity, when number of VMs does not exceed range 4...16.

With further increase of VMs amount, the time for distribution obtained by SAV algorithm grows within exponential dependence, whereas heuristic and genetic algorithms demonstrate linear growth of time.

The huge difference in work time of SAV algorithm and heuristic algorithms is explained by their nature. SAV is combinatorial algorithm when heuristic algorithms are just modification of sort algorithms.

In addition, an analysis of errors during work of algorithms was performed as proposed in [4]. A value of objective function ST, SM, SMGA, obtained as a result of heuristic algorithms work "T", "M" and manageable genetic algorithm

(MGA) respectively was compared to optimal solution provided by SAV. Figure 4 shows error of distribution of VMs plotted along ordinate axis, which shows how much more servers are used with distribution of VMs obtained by algorithms “T”, “M” and MGA, comparing to optimal allocation. The number of VMs requested to distribution on servers in DC is plotted along abscissa axis. Values of error are averaged from the results of 10 experiments.

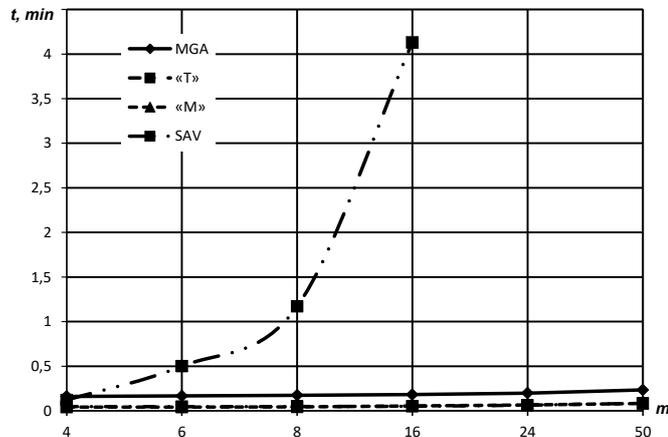


Fig. 3. Dependence of algorithms work time on problem complexity

The SAV algorithm gives slight gain in accuracy even on tasks with low complexity. With the increase in complexity, the error from heuristic and genetic algorithms grows, what indicates a dependence of error on the task complexity.

It should be noticed that for successful appliance of the SAV algorithm, it is necessary to have some amount of servers N_{\min} , for which it is certainly known that they can allocate all VMs from set K_j , $j = \overline{1, m}$. A value N_{\min} can be obtained with expert's help or with the use of one of heuristic algorithms.

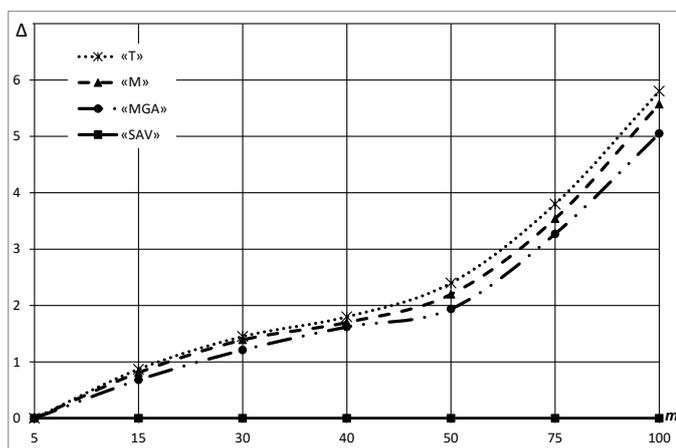


Fig. 4. Dependence of error of VMs distribution Δ relative to optimal solution depending on problem complexity

Researches have allowed to make the following recommendations for choosing an algorithm for VMs distribution on physical servers. For tasks where high accuracy

with small amount of VMs is the main priority, it is necessary to use the SAV algorithm because it finds optimal solution. The SAV can also be used in case where time for obtaining the result is not critical.

Heuristic and genetic algorithms make it possible to get a fast solution. Speed of work of such algorithms loosely depends on complexity of a task but the quality of solution is not on the same level as the solution of the SAV algorithm.

CONCLUSIONS

The result of researches made in this paper is an algorithm of discrete optimization based on SAV scheme for solving the problem of VMs allocation on physical servers of a data center. During development of algorithm, evaluations of time expenses of SAV algorithm were made, which depend on problem size. Comparative analysis of efficiency of different algorithms was done. It was proved that heuristic algorithms allow to obtain a solution really fast. With the use of such algorithms it is possible to reduce expenses for maintaining servers. The SAV algorithm requires significant time expenses but makes it possible to find the best solutions.

REFERENCES

- [1] J. H. Holland “Complex Adaptive Systems,” in *Daedalus*, vol. 121, no. 1, pp. 17–30; Winter 1992.
- [2] K. S. Leung, Q. H. Duan, Z. B. Xu and C. K. Wong, “A new model of simulated evolutionary computation: convergence analysis and specifications,” *IEEE Trans. on Evolutionary Computation*, vol. 5, no. 1, pp. 3–16, 2001.
- [3] S. Telenyk, E. Zharikov and O. Rolik, “Architecture and Conceptual Bases of Cloud IT Infrastructure Management,” in *Advances in Intelligent Systems and Computing*, Springer, 2017, pp. 41–62.
- [4] S. Telenyk, O. Rolik, P.S. Savchenko and M. E. Bodaniuk, “Manageable genetic algorithm in tasks of distribution of virtual machines in data centres,” *Visnyk of Cherkasy State Technological University*, vol. 2, pp. 104–113, 2011.
- [5] S. F. Telenik, A. I. Rolik, M. M. Bukasov and S. A. Androsov, “Genetic algorithms of decision of tasks of management resources and loading of centers of processing of data” *Automatic. Automation. Electrical engineering complexes and systems*, no. 1 (25), pp. 106–120, 2010.
- [6] S. Singh and I. Chana, “A Survey on Resource Scheduling in Cloud Computing: Issues and Challenges,” *Journal of Grid Computing*, pp. 1–48, 2016.
- [7] Z. Cao and S. Dong, “An energy-aware heuristic framework for virtual machine consolidation in cloud computing,” *The Journal of Supercomputing*, pp. 1–23, 2014.
- [8] M. Sun, W. Gu, X. Zhang, H. Shi, and W. Zhang, “A matrix transformation algorithm for virtual machine placement in cloud,” in *Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2013 12th IEEE Int. Conf. on. IEEE, 2013, pp. 1778–1783.
- [9] F. Lopez Pires and B. Baran, “Multi-objective virtual machine placement with service level agreement: A memetic algorithm approach,” in *Proc. of the 2013 IEEE/ACM 6th Int. Conf. on Utility and Cloud Computing*. IEEE Computer Society, 2013, pp. 203–210.
- [10] W. Wang, H. Chen, and X. Chen, “An availability-aware virtual machine placement approach for dynamic scaling of cloud applications,” in *Ubiquitous Intelligence & Computing and 9th Int. Conf. on Autonomic & Trusted Computing (UIC/ATC)*, 2012, pp. 509–516.
- [11] V. S. Mikhalevich, “Consecutive optimization algorithms and their application. I. II,” *Cybernetics*, no.1, pp. 45–55, no 2. pp. 85–88, 1965.
- [12] E. N. Sipko, “The method for sequential analysis of variants to solve a scheduling problem,” *Iskusstvenny intellekt*, no. 1, pp. 243–250, 2011.