

**FEATURES OF APPLICATION OF EVOLUTIONARY
ALGORITHMS FOR MULTI-PURPOSE OPTIMIZATION OF THE
OPERATION OF A VIRTUAL MACHINE IN A CLOUD SERVICE
ENVIRONMENT**

**Makarov A.E.¹ (Russian Federation), Varlamov A.A.² (United States of
America)**

Email: Makarov519@scientifictext.ru

*¹Makarov Anatoly Evgenievich - Solutions Architect,
IBM,
MOSCOW;*

*²Varlamov Aleksandr Aleksandrovich – Senior Solutions Architect,
LI9 INC,
NORTH CAROLINA, RALEIGH, UNITED STATES OF AMERICA*

Abstract: *the analysis of virtualization features of a cloud service's hardware complex is carried out, methods of distribution of virtual machines are considered. A methodology has been developed to reduce the problem of optimizing hardware resources to the mathematical problem of finding the minimum of a function. Analytical solutions are proposed to calculate the depreciation of the consumption of computing power, RAM, and electricity by the cloud service and minimize the delay in executing user requests. A generalized scheme of the methodology for constructing elite and non-elite evolutionary algorithms for multi-purpose optimization is developed.*

Keywords: *virtual machines placement, evolutionary algorithms, cloud service, computing power, random access memory, power consumption, latency, multiobjective optimization.*

**ОСОБЕННОСТИ ПРИМЕНЕНИЯ ЭВОЛЮЦИОННЫХ
АЛГОРИТМОВ ПРИ МНОГОЦЕЛЕВОЙ ОПТИМИЗАЦИИ РАБОТЫ
ВИРТУАЛЬНОЙ МАШИНЫ В СРЕДЕ ОБЛАЧНОГО СЕРВИСА**

**Макаров А.Е.¹ (Российская Федерация), Варламов А.А.² (Соединенные
Штаты Америки)**

*¹Макаров Анатолий Евгеньевич - архитектор решений,
IBM,
г. Москва;*

*²Варламов Александр Александрович – главный архитектор решений,
Li9 INC,
Северная Каролина, г. Райли, Соединенные Штаты Америки*

Аннотация: *проведен анализ особенностей виртуализации аппаратного комплекса облачного сервиса, рассмотрены методы распределения*

виртуальных машин. Разработана методология по сведению задачи оптимизации использования аппаратных ресурсов к математической задаче нахождения минимума функции. Предложены аналитические решения для расчета минимизации потребления облачным сервисом вычислительных мощностей, оперативной памяти и электроэнергии, а также по минимизации задержки при выполнении пользовательских запросов. Разработана обобщенная схема методологии построения элитарных и не элитарных эволюционных алгоритмов многоцелевой оптимизации.

Ключевые слова: *распределение виртуальных машин, эволюционные алгоритмы, облачный сервис, вычислительная мощность, оперативная память, энергопотребление, задержка, многоцелевая оптимизация.*

1. Введение

На сегодняшний день в области облачных вычислений значительную роль играет концепция виртуализации аппаратных ресурсов информационной системы, в частности, внедрение методов распределения виртуальных машин (virtual machine placement, VMP). Данную группу методов объединяет задание по автоматизации и оптимизации процесса контроля и использования ресурсов аппаратной платформы облачного сервиса, которые в общем случае могут быть классифицированы как (i) вычислительные мощности, (ii) оперативная память и (iii) информационное хранилище (рис. 1). Работа облачного сервиса включает в себя этап анализа запроса пользователя в соответствии с пунктами представленной классификации, а также этап распределения соответствующих ресурсов на уровне серверного комплекса. Так, вычислительные мощности сервиса определяются количеством центральных процессоров (ЦП) каждого из серверов, количеством ядер многоядерных ЦП, тактовой частотой и коэффициентом многоядерности ЦП; оперативная память — количеством элементов ОЗУ, их объемом памяти и быстродействием, информационное хранилище — количеством информационных накопителей, информационной емкостью каждого накопителя и скоростью передачи данных. Под оптимизацией в данном случае подразумевается построение такого алгоритма распределения аппаратных ресурсов, который позволит максимально снизить уровень потребления для всей системы, включая энергопотребление, а также даст возможность обработать больше пользовательских запросов и снизить задержку при их обработке [1, 2].

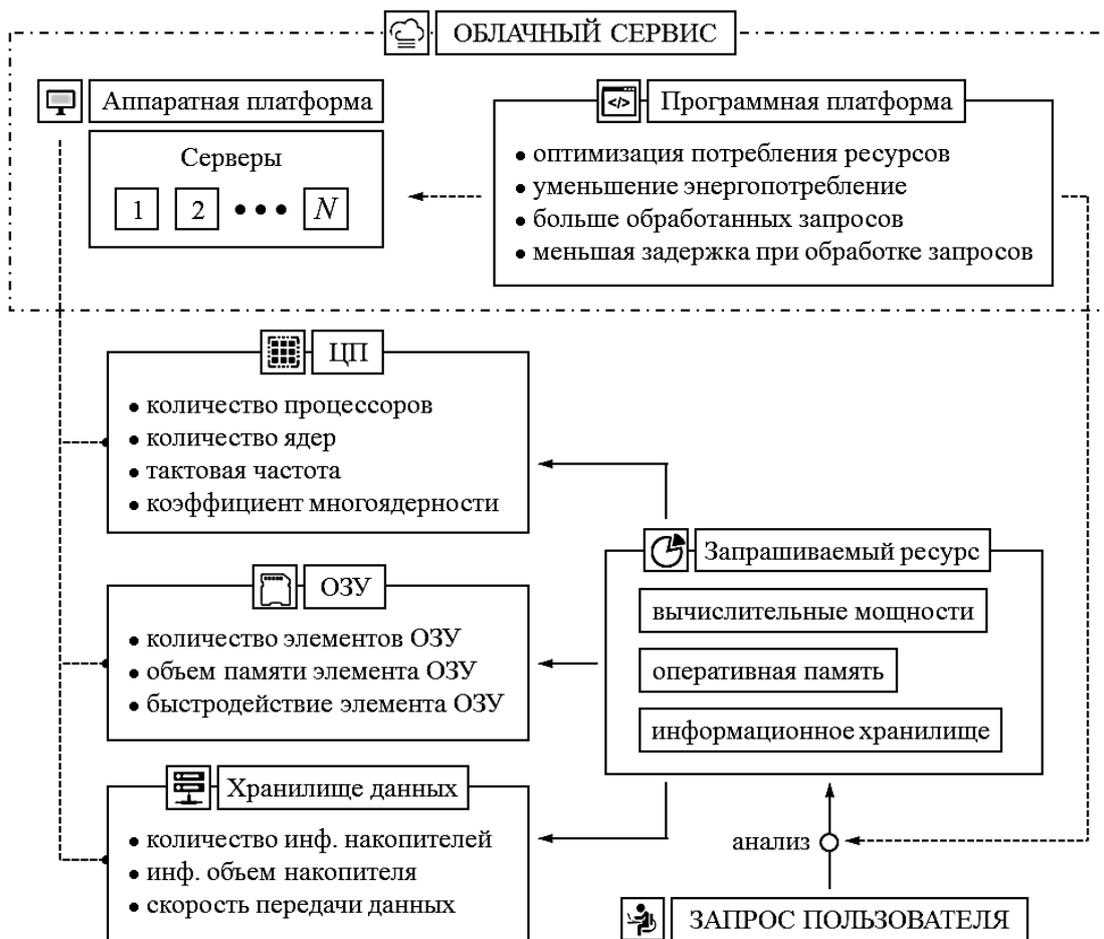


Рис. 1. Схема распределения виртуальных машин в рамках облачного сервиса

При этом, следует отметить, что эффективная организация процесса распределения виртуальных машин (virtual machine, VM) дает ряд дополнительных преимуществ для облачного сервиса. В частности, это простота настройки сети передачи данных, наличие расширенного инструментария масштабирования общей системы и подсистем, надежность хранения данных и функционирования системы при работе в автономном или полуавтономном режиме, что указывает на **актуальность данного исследования**.

Анализ последних исследований и публикаций в данной области указал на преимущества применения эволюционных алгоритмов (evolutionary algorithms, EA) при работе с многозадачной оптимизацией комплекса VM [3-5]. Среди основных преимуществ исследователи выделяют простоту реализации данного класса алгоритмов, которая связана с тем, что в EA не используется предварительная обработка входных данных, а также их адаптивность для широкого диапазона задач [3, 6]. Методика организации VMP-процедуры, которая базируется на применении EA и стохастическом

популяционном поиске (population-based search, PBS) позволяет разработать стратегию параллельного поиска и, соответственно получить на выходе мультимодальное решение, а также нормализовать выходные значения с целью уменьшения вероятности отклонений от элементов множества вариантов пространства решений [7, 8]. **Нерешенной частью общей задачи исследования** является высокая нагрузка ЕА на аппаратные ресурсы серверов (physical machine, PM) облачного сервиса, что в ряде случаев является ограничением для широкого применения указанных алгоритмов [9-15].

Таким образом, **целью работы** стало построение целостной методологии разработки моделей ЕА, которая позволяет оптимизировать данный класс алгоритмов в соответствии с особенностями задачи по реализации процедуры VMP облачного сервиса.

2. Методология исследования

Рассмотрим модель комплекса виртуальных машин как набор из $k \in [1; K]$ элементов, множество которых разбито на $n \in [1; N]$ доменов (n), каждый из которых регулируется соответствующим доменным контролером (domain controller, DC). Для построения математического аппарата необходимо ввести следующие логические переменные:

- $b_1(n, k)$, значение $b_1(n, k) = 1$ если виртуальная машина VM_n принадлежит домену DC_k , и $b_1(n, k) = 0$ — в противоположном случае;
- $b_2(k)$, значение $b_2(k) = 1$ если доменный контролер DC_k подключен и $b_2(k) = 0$ — в противоположном случае.

Коэффициент $C_\Sigma(k)$ определяет общий уровень использования ресурсов (Resource Wastage) в рамках домена k , и рассчитывается на основе коэффициентов запроса на вычислительную мощность $C_C^\Sigma(k)$ и оперативную память домена $C_M^\Sigma(k)$, а также граничные значения для соответствующих ресурсов домена ($C_C^{max}(k)$ и $C_M^{max}(k)$):

$$C_\Sigma(k) = \frac{b_2(k) \cdot |(C_C^{max}(k) - C_C^\Sigma(k)) - (C_M^{max}(k) - C_M^\Sigma(k))|}{C_C^\Sigma(k) + C_M^\Sigma(k)}, \quad (1)$$

где C_C^Σ и C_M^Σ определяются как

$$\begin{cases} C_C^\Sigma(k) = \sum_{n=1}^N (b_1(n, k) \cdot C_C(n)) \\ C_M^\Sigma(k) = \sum_{n=1}^N (b_1(n, k) \cdot C_M(n)) \end{cases} . \quad (2)$$

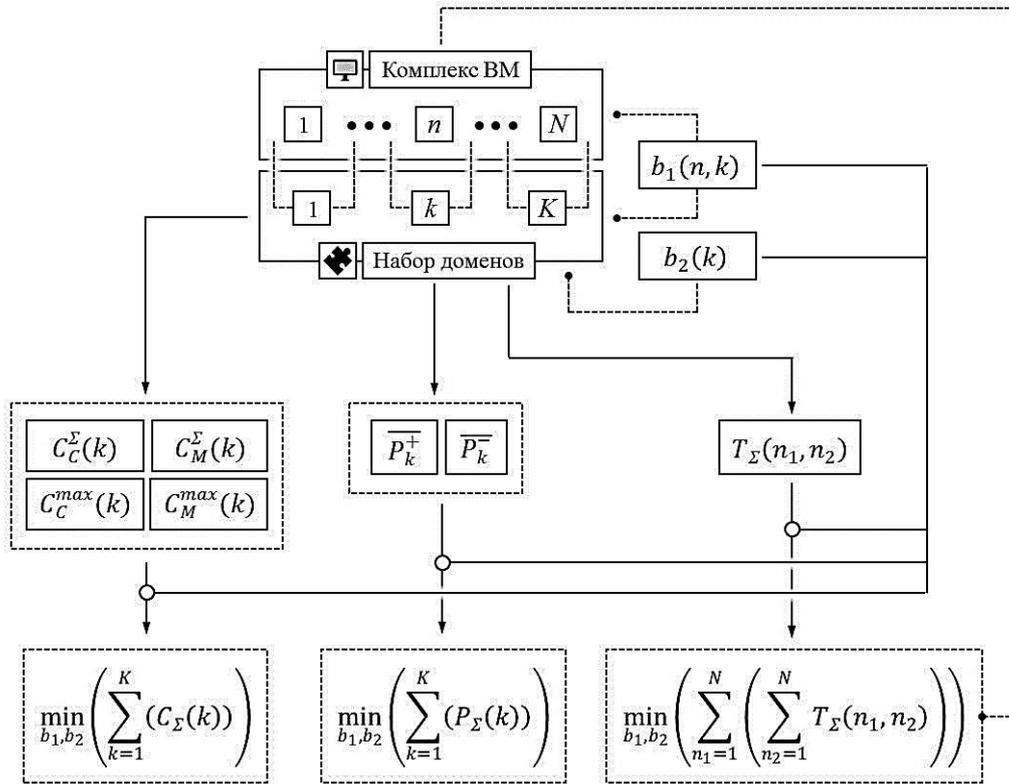


Рис. 2. Схема минимизации нагрузки на аппаратный ресурс комплекса виртуальных машин облачного сервиса

Соответственно, проблема оптимизации использования вычислительных ресурсов и ресурса оперативной памяти аппаратно-программной платформы облачного сервиса может быть сведена к задаче нахождения минимума суммы значений $C_{\Sigma}(k)$ для всех $k \in [1; K]$:

$$C_{\Sigma} = \min_{b_1, b_2} \left(\sum_{k=1}^K (C_{\Sigma}(k)) \right). \quad (3)$$

Аналогично можно на математическом уровне решить проблему минимизации потребления электроэнергии инфраструктурой облачного сервиса (рис. 2):

$$P_{\Sigma}(k) = b_2(k) \cdot \left((\overline{P}_k^+ - \overline{P}_k^-) \cdot C_C^{\Sigma}(k) + \overline{P}_k^- \right), \quad (4)$$

где \overline{P}_k^+ — показатель среднего значения потребляемой мощности используемым доменом, \overline{P}_k^- — показатель среднего значения потребляемой мощности неиспользуемым доменом. Соответственно, минимальное значение рассчитывается как:

$$P_{\Sigma} = \min_{b_1, b_2} \left(\sum_{k=1}^K (P_{\Sigma}(k)) \right). \quad (5)$$

Наконец, следует минимизировать задержку, которая происходит в системе виртуальных машин при обмене данными:

$$T_{\Sigma} = \min_{b_1, b_2} \left(\sum_{n_1=1}^N \left(\sum_{n_2=1}^N T_{\Sigma}(n_1, n_2) \right) \right), \quad (6)$$

где функция $T_{\Sigma}(n_1, n_2)$ определяет задержку в передаче данных между виртуальными машинами VM_{n_1} и VM_{n_2} .

3. Разработка методологии построения эволюционных алгоритмов многоцелевой оптимизации

Рассмотрим набор из I решений (solution set, SS) задачи оптимизации процедуры VMP облачного сервиса в виде множества $S = [S_1, S_2, \dots, S_i, \dots, S_I]$. В рамках решения задачи многоцелевой оптимизации, где количество целей составляет J , каждый из элементов множества может быть представлен как $S_i = [D_i^1, D_i^2, \dots, D_i^j, \dots, D_i^J]$, каждый из элементов подмножества (D_i^j для $\forall i \in [1; I]$ и $\forall j \in [1; J]$) представляет собой размерность задачи. Оптимальное решение задачи оптимизации для S_i , таким образом, может быть получено при помощи векторов $\vec{F}_i = [F_i^1, F_i^2, \dots, F_i^j, \dots, F_i^J]$, который удовлетворяет ограничения, накладываемые начальными условиями поставленной задачи, и дает возможность ранжировать конечные решения на основе набора целевых функций $Z(\vec{F}_i) = [Z_1(\vec{F}_i), Z_2(\vec{F}_i), \dots, Z_h(\vec{F}_i), \dots, Z_H(\vec{F}_i)]$ через построение зависимостей типа $\vec{F}_x < \vec{F}_y$ и $\vec{F}_z \preceq \vec{F}_y$ (рис. 3).

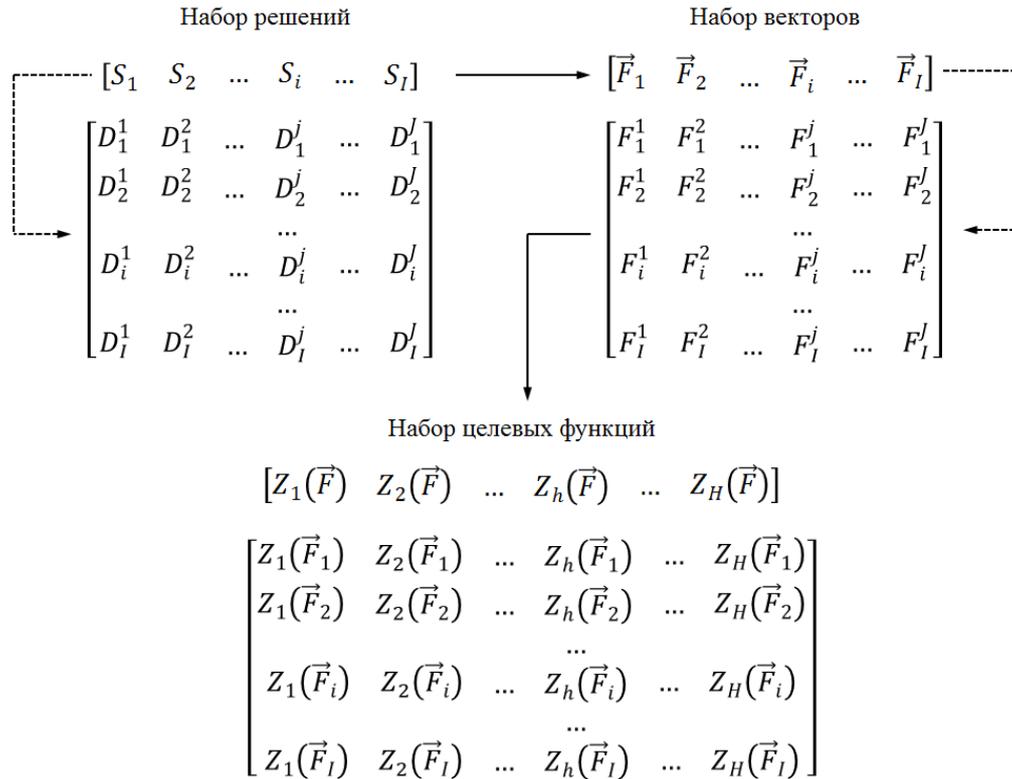


Рис. 3. Алгоритм многоцелевой оптимизации процедуры VMP

Проводя классификацию многоцелевых эволюционных алгоритмов (multiobjective evolutionary algorithms, MOEA), которые могут быть использованы при решении задачи оптимизации процедуры VMP, следует выделить две группы:

- элитарные MOEA (elitist multiobjective evolutionary algorithms, E-MOEA), среди которых выделяются избранные элементы множества, что позволяет поддерживать уровень производительности алгоритма;
- неэлитарные MOEA (non-elitist multiobjective evolutionary algorithms, NE-MOEA).

В рамках предложенной математической модели из числа NE-MOEA можно использовать такие алгоритмы как генетический алгоритм с векторной оценкой Шаффера (Schaffer's vector-evaluated genetic algorithm, VEGA), многоцелевой генетический алгоритм (multiobjective genetic algorithm, MOGA), а из числа E-MOEA: генетический алгоритм с недоминируемой сортировкой (non-dominated sorting genetic algorithm, NSGA), эволюционный алгоритм на основе силы Парето (strength pareto-evolutionary algorithm, SPEA) и многоцелевой эволюционный декомпозиционный алгоритм (multiobjective evolutionary algorithm/decomposition, MOEA/D).

Выводы

Проведенный анализ ключевых подходов по виртуализации аппаратного комплекса облачного сервиса и, в частности, методов распределения виртуальных машин инфраструктуры облачного сервиса показал преимущества эволюционных алгоритмов. Был предложен и разработан подход по сведению задачи оптимизации использования аппаратных ресурсов аппаратной платформы комплекса виртуальных машин к математической задаче нахождения минимумов целевых функций. Полученные в результате исследования аналитические решения для расчета минимизации потребления облачным сервисом вычислительных мощностей, оперативной памяти и электроэнергии, а также по минимизации задержки при выполнении пользовательских запросов могут быть далее использованы при работе с широким классом практических задач. На их основе также была разработана обобщенная схема методологии построения элитарных и неэлитарных эволюционных алгоритмов многоцелевой оптимизации.

Список литературы / References

1. *Dimitrov V.*, 2016. Cloud Programming Models (MapReduce). Encyclopedia of Cloud Computing, 596-608. doi:10.1002/9781118821930.ch49.
2. *Gordon D.G.*, 2016. Legal Aspects of Cloud Computing. Encyclopedia of Cloud Computing, 462-475. doi:10.1002/9781118821930.ch38.
3. *Nedjah N., De M.M. & Lopes H.S.*, 2018. Evolutionary multi-objective system design: Theory and applications. Boca Raton, FL: CRC Press, Taylor & Francis Group.
4. *Ishibuchi H. & Shibata Y.* (2004). Single-Objective And Multi-Objective Evolutionary Flowshop Scheduling. Applications of Multi-Objective Evolutionary Algorithms Advances in Natural Computation, 529-554. doi:10.1142/9789812567796_0022.
5. *Fei Z., Li B., Yang S., Xing C., Chen H., Hanzo L.*: A survey of multi-objective optimization in wireless sensor networks: metrics, algorithms, and open problems. IEEE Commun. Surv. Tutor. 19 (1), 550–586 (2016).
6. *Pan J.* (2020). Genetic and evolutionary computing: Proceedings of the thirteenth International Conference on Genetic and Evolutionary Computing, November 1-3, 2019, Qingdao, China. Gateway East, Singapore: Springer.
7. *Branke J., Deb K., Miettinen K., Slowiński R.*: Multiobjective Optimization, Interactive and Evolutionary Approaches. Springer, Berlin, Heidelberg (2008).
8. *Baykasoğlu A. & Ozsoydan F.B.* (2017). Evolutionary and population-based methods versus constructive search strategies in dynamic combinatorial optimization. Information Sciences, 420, 159-183. doi:10.1016/j.ins.2017.08.058.

9. *Jansen T.* (2012). Evolutionary Algorithms and Other Randomized Search Heuristics. Analyzing Evolutionary Algorithms Natural Computing Series, 7-29. doi:10.1007/978-3-642-17339-4_2.
10. *Song B. & Li V.O.* (2020). A Revisit of Infinite Population Models for Evolutionary Algorithms on Continuous Optimization Problems. Evolutionary Computation, 28 (1), 55-85. doi:10.1162/evco_a_00249.
11. *Zhang Q., Li H.:* MOEA/D: a multiobjective evolutionary algorithm based on decomposition. IEEE Trans. Evol. Comput. 11 (6), 712–731 (2007).
12. *Deb K. et al.:* A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6 (2), 182–197 (2002).
13. Multiobjective Genetic Algorithm for Pareto-Optimal Route Selection. (2009). Intelligent Engineering Systems through Artificial Neural Networks, 323-330. doi:10.1115/1.802953.paper41.
14. *Vandeva E.* (2012). MultiObjective Genetic Modified Algorithm (MOGMA). Cybernetics and Information Technologies, 12 (2), 23-33. doi:10.2478/cait-2012-0010.
15. *Kamkar I. & Akbarzadeh T.M.* (2010). Multiobjective cellular genetic algorithm with adaptive fuzzy fitness granulation. 2010 IEEE International Conference on Systems, Man and Cybernetics. doi:10.1109/icsmc.2010.5642401.