Data Driven Meta-Heuristic-Assisted Approach for Placement of Standard IT Enterprise Systems in Hybrid-Cloud

Andrey Kharitonov, Abdulrahman Nahhas, Hendrik Müller and Klaus Turowski
Faculty of Computer Science, Otto von Guericke University, Magdeburg, Germany

Keywords: Commercial-off-the-Shelf Enterprise Applications, Capacity Management as a Service, Hybrid-Cloud.

Abstract: We address the problem of hybrid-cloud placement selection for commercial off-the-shelf IT enterprise applications with the sizing done based on workload profiles collected from real-world production systems. The proposed approach leverages techniques based on evolutionary meta-heuristics with a multi-criteria weighted sum objective function. A placement decision is made between an on-premises data center and a public cloud, using real pricing information for virtual machines, storage, and networking published by the public cloud vendor via automation APIs and on-premises cost estimation as a share of expense per service. Additional objectives, such as expertise and non-functional requirements, are encoded in a numerical form for the objective function. The evaluation is performed as single and multi-objective optimization by employing genetic algorithm, and non-dominated-sorting genetic-algorithm-III on the case study of an SAP landscape hybrid-cloud placement on a selected public cloud with real workload data collected during day-to-day business operations, indicating the viability of the approach.

1 INTRODUCTION

Standard IT enterprise systems encompass software solutions designed to fulfill the needs of varying size organizations (i.e., enterprise resource planning (ERP) systems). Specifically, in medium and large enterprises, such systems are often comprised of multiple semi-independent sub-systems, such as a variety of application systems and databases. Infrastructure refresh, expansion, or transformation involves a decision-making process on the software systems placement within a heterogeneous infrastructure consisting of private data centers and public cloud placement options (Missbach et al., 2016).

When simply selecting the cheapest available infrastructure fulfilling the required key performance indicators (KPI) configuration isn't enough, other requirements and relevant indicators must be considered in the decision-making process. Furthermore, the efficiency of applying this process in real life is of utmost importance. It is also important to avoid developing a process that resembles a black box for the user expert or a customer receiving placement recommendations generated by this process.

Therefore, the motivation for this work is a need to develop an easily explainable process that can be delivered as a service providing support in a hybrid-cloud transformation process of standard IT enterprise applications, with SAP-based IT infrastructure as a particular case study. The service is to provide data-driven decision support on the challenge of identifying the best-suitable operating environments for given SAP workloads. Based on an arbitrary number of customer-specific quantifiable requirements, existing constraints, measured workloads, resource demands, and cost estimations, a placement recommendation is provided for each assessed SAP system (identified by its system ID, or, in short, SID). In principle, this is a multi-criteria decision problem, which can be tackled through appropriate approaches (i.e., weighted sum model), also in the area of cloud provider selection(Chauhan et al., 2020).

The search space of possible target environments for each SID is formed by the variety of available placement configurations and regions provided by a public cloud provider and, if applicable, the customer's on-premises data center. In order to measure what is "best-suitable", each potential solution is associated with a score that represents its fulfillment degree of customer requirements, constraints, and the resulting costs. Therefore, it is a challenge to identify solutions that maximize the score.

Due to the complexity of the problem, multiple solution search approaches for similar problems were

Kharitonov, A., Nahhas, A., Müller, H. and Turowski, K.
Data Driven Meta-Heuristic-Assisted Approach for Placement of Standard IT Enterprise Systems in Hybrid-Cloud.
DOI: 10.5220/0011726600003488
In *Proceedings of the 13th International Conference on Cloud Computing and Services Science (CLOSER 2023)*, pages 139-146
ISBN: 978-989-758-650-7; ISSN: 2184-5042

proposed in the existing literature. Including those based on algorithms that belong to the class of meta-heuristics (Rahimi et al., 2022), which we also adopt within this work. Furthermore, in the presented case study, of Microsoft Azure, infrastructure costs are calculated based on publicly available list prices for compute, network, and storage components per system component placement. In the case of on-premises environments, a generic data center cost model is used and parameterized with customer input to estimate cost summary for system components. In both cases (cloud and on-premises), required components are sized based on real customer workloads and resource demands. For this purpose, a workload data collection software measures resource demands and capacity for the defined set of SAP systems over a defined period of time. The impact of using the workload collector software on the overall system performance is negligible.

Essentially, the goal of the approach is to answer the following questions posed by a customer undergoing SAP landscape transformation:

Q1: Which of my SAP workloads are suitable for cloud operations, and which are suitable for on-premises?

Q2: What would be the infrastructure costs to operate my SAP workloads in a public cloud?

Q3: What's the most cost-effective distribution of my SAP workloads across on-premises infrastructure and public cloud regional data centers?

Q4: What's the right level of computational resources at which point in time?

Q5: How does Remote Function Call (RFC) network traffic between my SAP instances affects pricing?

2 DATA DRIVEN SIZING

Redesign or transformation of the hybrid-cloud infrastructure capacities for the fulfillment of the KPIs can be done based on the real-world workload profiles. As mentioned in the previous section, this workload profile can be recorded and used for analysis or alternatively, performance unit tests can be utilized on the real system (Horký et al., 2015). In this work, we concentrate on making use of such workload profiles for standard off-the-shelf enterprise systems using SAP as a specific use case, well suitable for data-driven sizing capacity (Müller et al., 2022).

Alongside the KPIs, pricing plays a crucial role in planning the hybrid-cloud infrastructure as it's necessary to find the balance between the costs of system placements in private data centers and pricing plans in considered public clouds. Estimating the costs of private data centers is not trivial (Patel and Shah, 2005), especially in cases of hybrid-cloud component distribution of the same system. In such cases we can rely on system component-based cost estimations (Brogi et al., 2019), while relying upon the workload profiles (CPU, memory, storage, network utilization). Similarly, we can rely on the workload profiles to select an optimal pricing model (Wu et al., 2019) per system per cloud placement within a specified planning horizon, with automation facilitated via integration with the pricing APIs of the public cloud providers.

Pre-calculation of all viable placement combinations for a landscape consisting of hundreds of systems and databases is computationally and memory expensive. For each pricing component, which depends on the landscape configuration, a matrix with dimensions $|K| * |P| \times |K| * |P|$ would have to be calculated, where K is a set of systems in the landscape, and P is a set of viable individual placement configurations. Instead, such costs, which depend on the configuration of the landscape, are estimated as part of the target function of a chosen meta-heuristic during the evaluation of a solution candidate. It is important to note that the placement decision can not be made for each system component in isolation as existing interdependencies (e.g., network, storage) affect not only KPIs but also the pricing. Such interdependencies can be inferred from the corresponding workload profile.

3 DECISION OBJECTIVES

The problem discussed in this work is the multi-attribute decision-making problem (Zanakis et al., 1998), which we address with an approach similar to a weighted sum for optimization method (Marler and Arora, 2010). That means that we consider multiple objectives, which might have a conflicting nature. The simplest example of such conflicting objectives is the maximization of the infrastructure capacities that stands in direct opposition to price minimization. In the real world, the decision objectives are more numerous and intricate. Some of these objectives are universal from project to project, but some can be unique functional or non-functional business requirements. Such objectives would be defined and quantified for each transformation or infrastructure upgrade project individually. Therefore, it's important to have a mechanism that supports the addition of an arbitrary number of new objectives in an intuitive way.

3.1 Objectives Definition

To illustrate this principle, in Table 1 we provide an example list of such additional objective categories with values and weights that were considered within the context of this work. These objectives were defined during a systematic interviewing process of experts in the field of SAP infrastructure architecture as well as the customers undergoing a refresh process of SAP infrastructure.

We asked the interviewees to provide a tuple of three numeric values for each requirement $\langle v, w^{onprem}, w^{cloud} \rangle$. The first value v denotes the client's answer to the stated requirement importance or an assessment question in a numerical representation. The values v, in principle, can be denoted on any numerical scale that is logically sound or convenient for any given requirement, but the upper and lower bounds of the scale must be known for later normalization. This means that in the end, every requirement is quantified in the range of $[0,1]$, which is selected for simplicity of calculation as any arbitrary numerical scale can be easily normalized to this range. If a requirement can't be represented as a numerical scale, it can't be used in the proposed approach directly and must be either considered in a post-processing logic or decomposed to simple quantifiable elements.

Values w^{onprem} and w^{cloud} represent the weights in the range $[0,1]$, which denote the importance of the requirement for the on-premises system placement solution and a public cloud placement solution, respectively. This means that every objective is associated with separate importance weights in the range $[0,1]$ separately for on-premises and cloud. The weights are applied via the means of simple multiplication during the assessment phase of the system placement to the value v. While in our short example, we operate with two types of weights, this principle can be intuitively extended to vendor-specific clouds or different types of private data centers.

The provided in Table 1 short example of such possible objectives consists of a total of four categories, differentiated by the designation of two-letter prefixes. Weights for the specific placement type within a single category are expected to be always summed up to 1. Note that the provided values for the competencies favor the on-premises operations because the values are based on the customer and expert interviewees. Every category is also associated with its own weight, as denoted in Table 2. Important to note that these criteria are high-level and not meant to be representative of real-world project complexity.

In the example above, even though values for all objectives are presented to the experts and the customers on a scale with a range $[0,10]$, the intuition behind these values differs from category to category. Values v in "Expertise" determine the level of expertise that is already present within the organization. These values in the categories "Non-functional requirements" and "Business strategy" determine the importance of the objectives for the customer according the specifics of business goals and strategies of the customer. These values are weighted and considered directly in the decision process.

All of the costs-related values v are subject to additional processing and not considered directly in the decision process. Instead, the value v denotes an additional level of importance for the specific cost component and acts as a scaling factor similar to weights but independent from on-premises or cloud placement type. While in the example above, such value is relatively redundant, as well as the explicit separation of costs between compute and networking, this demonstrates how such approach can also serve as a way to precisely balance cost impact on the decision process for more complex pricing models. These costs for compute and storage are precalculated as described in Section 2 for every possible placement type for each systems present in the IT landscape and added to the set denoted P^{γ}. Pricing in the set is defined per system $P^{\gamma_{sys}} \in P^{\gamma}$ All prices are normalized to values in the range $[0,1]$ and subtracted from 1 to enable the values to be used as part of the maximization problem: $\forall p \in P^{\gamma_{sys}}.P_p = 1 - \left(P_p^{\gamma_{sys}} - min\left(P^{\gamma_{sys}}\right)\right) \div \left(max\left(P^{\gamma_{sys}}\right) - min\left(P^{\gamma_{sys}}\right)\right)$, where P is a final set of normalized prices $P_p \in P$. Estimations for additional pricing points (i.e., networking, backups) are processed in a similar manner but normalized based on the landscape-wide minimum and maximum values.

3.2 Constraints

Not every technically possible solution that satisfies the business requirements mentioned above is, in fact, viable. For example, if a company processes sensitive information (e.g., personal, medical, commercial) it falls under certain regulations concerning data placement and processing (Hippelainen et al., 2017; Sarferaz, 2022). Therefore, if we take any public cloud provider with data centers located all over the world, only a subset of these data centers and combinations of offerings within others will be viable for consideration. For example, a German company that falls under such regulations (i.e., GDPR, DSGVO), might want to only consider data centers within the European Union. This is a hard constraint that cannot be broken as then the company risks non-compliance legal cases raised against it.

Table 1: Example objective values.

Designation	Description	v	w^{cloud}	w^{onprem}
CO 1	Compute and storage costs	8	0.5	0.5
CO 2	Networking costs	8	0.5	0.5
EX 1	Setting up on-prem SAP landscapes	10	0	0.5
EX 2	Setting up cloud SAP landscapes	5	0.25	0
EX 3	Maintaining on-prem SAP landscapes	10	0	0.5
EX 4	Maintaining cloud SAP landscapes	5	0.25	0
EX 5	Cloud Operations	5	0.25	0
EX 6	Cloud Architecture	4	0.25	0
NF 1	Performance	8	0.2	0.8
NF 2	Scalability & elasticity	5	0.4	0.1
NF 3	Innovation degree	6	0.4	0.1
ST 1	Level of control	5	0.1	0.4
ST 2	Carbon Footprint	2	0.1	0.2
ST 3	OPEX importance	7	0.7	0.1
ST 4	IT impact on business model	8	0.1	0.3

Table 2: Example objective categories.

Prefix	Objective category	Weight
CO	Costs	0,35
EX	Expertise	0,25
NF	Non-functional requirement	0,25
ST	Business strategy	0,15

Such constraints can be included as part of the objectives and processed as part of the objective function calculation. However, in this particular case, it would cause needlessly wasteful use of the computational resources as placement hard constraints can easily be satisfied by simple preliminary filtering of the possible placement options before any calculation takes place. In the aforementioned example, the placement options are not a whole set of all possible public cloud provider locations but a subset limited to the region of the European Union.

Another type of constraint is a soft constraint. Soft constraint reflects a strong preference that can be overturned under the weight of the other objective values. The most obvious examples of such constraints are co-location and anti-co-locations (Jammal, M. et al., 2015) for different systems within the IT landscape. For example, there can be two or more systems that, due to technical requirements, such as a high amount of data transferred between them, are preferred to be placed together. Alternatively, two systems that are individually critical to the business processes and the loss of both of them at the same time can impact business processes, and therefore, these systems preferably shouldn't be placed together. Note that the latter example can also be mitigated by employing high-availability options on-premises and in the cloud, but these result in significant cost increases, which might not be warranted by the business significance of the system. High-availability options, where needed, are selected as part of the preconfiguration of the placement for the specific system. Soft constraints are associated with a weight, which determines the significance of the constraint to the overall assessment of the selected placement.

3.3 Objective Function

We approach the problem of placement decision in two variants, which can be referred to as follows: multi-objective solved as mono-objective, pure multi-objective (Pires and Baran, 2015). Or termed alternatively, the problem is represented as a single-objective and as a many-objective (Helbig and Engelbrecht, 2013). Furthermore, since the discussed objectives must be represented in a numerical way suitable for usage in an automation process of calculation of best available options from the presented alternatives and for different questions ranges of values can be different, normalization is performed to bring all objectives to a common scale. Considering the practical orientation of the presented approach, simplicity of the objective function was one of the priorities as it must be easily explainable and understandable on all levels of decision-making process in real-world enterprises.

Objectives are grouped into a set of categories denoted C. Each $c \in C$ is associated with a weight $w_c \in W$, and every weight is with placement type, such as "on-premises" or "cloud" in the use case discussed in this work: $W = W^{onprem} \bigcup W^{cloud}$. All weights wights within the category per placement are summing up to exactly to 1. Meaning that for every $c \in C$, on-premises sum of weights is $\sum W_c^{onprem} = 1.0$

and cloud is $\sum W_c^{cloud} = 1.0$.

$$\varsigma(c) = \frac{\sum_{x \in c} \zeta(x_v) * x_w}{|c|} \quad (1)$$

$$\zeta(x) = \begin{cases} x, x \in \mathbb{R} \\ \varsigma(x), x \neq \emptyset \end{cases} \quad (2)$$

$$S_p = \sum_{c \in C_p} \varsigma(c) * W_c^p \quad (3)$$

Finally, the overall weighted score is denoted S_p, as presented in Equation 3, with a recursive system of Equation 1 and Equation 2. The approach taken in this work is inspired by using the weighted sum model(Chauhan et al., 2020) as a basis and modifying it to better suit the problem discussed in this work. Through this simple, easily explainable calculation, we receive a numerical representation that includes an arbitrary number of functional and non-functional requirements coupled with the pricing components. This representation theoretically allows a hierarchical composition of the requirement representation where each value in each category can be represented either as a real number $\mathbb{R}$ or as a subset of values with its own weights. However, because every subset, when weighted and summed up, is normalized to the range $[0,1]$, the further down below in the hierarchy the value is located, the less relevant these values become to the final score.

Therefore, the possible maxim score for system placement configuration is simply a total number of all top-level categories: $S_p^{max} = |\{c \mid c \in C \wedge \{r \mid r \in C \wedge c \in r\} = \emptyset\}|$. Then the minimum score is simply $S_p^{min} = 0$. The final scoring for the entire landscape of multiple systems placed and configured at available locations is simply the sum of scores acquired for all these systems. This final summation of scores can be directly used for the direct comparison of different landscape placement configurations and the selection of the most suitable one according to the price and all of the specified additional requirements. In this case, the higher the score is, the better. This approach can be directly used as an objective function in a variety of heuristic or meta-heuristic algorithms as a score maximization problem.

However, there is a variety of algorithms that are suitable for solution search in a multi-objective fashion. In this case the described above mechanism can still be applied, but instead of calculating of the final score, we operate with the top-level requirement categories as multiple objectives directly with the according weights without summing them up. We simply select all categories $C^{\Delta} \subseteq C$ that aren't child to any other, as described in Equation 4 and then apply appropriate weights as seen in Equation 5.

$$C^{\Delta} = \{c \mid c \in C \wedge \{r \mid r \in C \wedge c \in r\} = \varnothing\} \quad (4)$$

$$\varphi = \left\{ \varsigma(x) * W_x \mid x \in C^{\Delta} \right\} \quad (5)$$

Furthermore, the mentioned above scores in the Equation 3 and Equation 5 are calculated per system in the landscape. For the landscape-wide final score, a sum of the scores is taken after the placement locations are selected $P^{\Delta} \in P$ and systems are assigned to these placements P_K^{Δ} and $K_p = \{ k_p \mid k \in K \wedge k \in P^{\Delta} \}$.

Constraints, if defined, are calculated based on the entire landscape placement. For each individual soft constraint, a percentage of satisfaction is calculated and then normalized to the range $[0,1]$. After all of the constraint satisfaction values are calculated for each constraint type, a weighted average of these values is calculated and used for further assessment together with the scoring function. This resulted value is used as a multiplicator for the calculated score in a single-objective variate, effectively scaling the resulted score by the percentage of the soft constraint satisfaction. In the multi-objective variant, this value becomes just an additional objective.

4 OPTIMAL PLACEMENT

Capacity planning and placement of the landscape that consists of just a couple of systems with pre-calculated sizing and costs for the compute and storage components of the VMs and just a few possible placement options is a straightforward problem that can be solved analytically. However, decision-making becomes more complicated on more complex landscapes with a variety of different placement options and requirements, which falls into a category of problems that is, in fact, an NP-hard problem (Bichler et al., 2006). The total number of solutions depends on the number of placement options in set P and the services in set K and equates to $|P|^{|K|}$ (Hyser et al., 2007).

To simplify and speed up the process of decision-making, we can employ meta-heuristic optimization algorithms that are known to work well with objective maximization problems based on objective values similar to those described in the previous section. As an example and to validate the scoring function, we employ a classic meta-heuristic genetic algorithm (GA), which is traditionally employed for solving optimization problems (Back, 1996) for a single objective function and a Non-Dominated-Sorting-Genetic-Algorithm-III or, in short, NSGA-III (Deb and Jain,

2014), that is well suitable for the multi-objective optimizations.

We select the classic genetic algorithm metaheuristic for the single objective problem solution due to its simplicity and easy-to-understand and implement principle, which is inspired by evolutionary theory. GA metaheuristic is a technique for a solution search in a problem search space within a number of generations, directed by a so-called fitness function that determines the quality of the solution. Solution candidates in this class of algorithms are named individuals. A large number of GA variations exist, but in principle, they share the same three main functions: initialization, mutation, crossover, and selection. Most of the hyperparameters are also common across many GA variations (i.e., population size, number of evolutionary generations, crossover rate).

We encode individuals, or solution candidates, as a sequence of whole numbers with a length of $|K|$, where K is a set of systems in the landscape. Each number in the sequence has a range of $(0, |P|]$ where each number corresponds to an index of a possible placement configuration specific to the system $p_k \in P_k$. In our encoding, every system is allowed to be placed on every considered on-premises or cloud location. If a specific placement isn't valid for the specific system, we invalidate the entire individual. Every individual is then evaluated directly with a single-objective function discussed in subsection 3.3.

For multi-objective solution search, we encode a solution individual in the same way, but the evaluation function used is now based on Equation 5, and every value is considered as a separate objective. In our example based on NSGA-III, all of the objective values are considered equally important by the metaheuristic itself, but the category weights mentioned in subsection 3.1, are applied as part of the aforementioned equation.

The key parameters in the family of genetic algorithm meta-heuristics are common for all algorithms. First is the maximum number of generations, denoted *maxGen*, controlling the number of iterations the genetic algorithm goes through. Next is the size of the population, denoted pop, which controls the number of solution individuals within every generation. Solution individual mutation probability, $prob_{mut}$, which determines the probability of random changes in the solution individual. And finally, λ, the total number of children produced during recombination (partially combining different solution individuals together). Many different approaches exist for the selection of the best individual within the generated population (Kruse et al., 2011), but in this work, we rely specifically on tournament selection.

5 EVALUATION

The presented approach of answering the five questions stated in section 1 is evaluated with a real-world use case based on a partial cloud transformation of SAP landscape with target placement environments located on-premises and in the public cloud infrastructure of Microsoft Azure. We consider an example IT landscape consisting of nine SAP systems, each identified by a System ID (SID), some of which are independent of others, and some have Remote Function Call (RFC) network connections recorded in the workload profiling data. These interconnections are illustrated in Figure 1. The volume of data transfer is taken as an average per hour within the recorded month-long workload profile, and this value is used for network cost estimations.

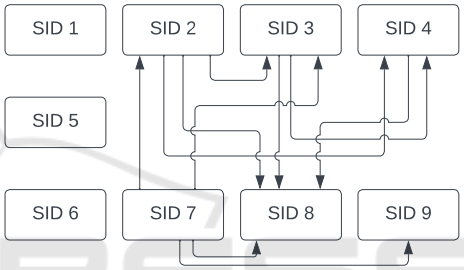

Figure 1: Validation example network connections.

Costs for on-premises and public cloud placement options are estimated for the horizon of planning of five years. Placement options are generated per system component and based on the recorded workload profiles and are viable capacity-wise. No potential workload profile changes or pricing changes are considered. For cloud-based placements, only the data centers within the European Union were considered. In total, 38 cloud placement configuration permutations per SID distributed across 19 locations were considered. Only SAP-certified Azure VMs were considered. For on-premises, we have considered a single location with the capacity limit, which we use as a basis for on-premise cost calculation formed by the measured workload plus a parameterized buffer of 35% for both CPU and main memory.

Pricing for compute resources and adequate size of the appropriate type of storage for SAP systems at the given horizon is estimated, without considering networking costs for cloud placements, to be on average 16% cheaper for eight out of nine systems placed on-premises in comparison to the same systems in cloud placements, including electricity and cost share per system for additional expenses (i.e., facility maintenance, IT staff, networking equipment). The only

system with the estimated running costs cheaper on the cloud is SID 4, with on-premise placements being significantly cheaper, 29% on average.

We investigate two use cases, with and without constraints. If constraints are present, these are introduced to promote the optimization algorithm to find a solution that is oriented toward reducing networking costs faster by co-locating communicating systems. Co-location constraints are defined in pairs of systems with a weight of 1.0.

Execution of both single- and multi-objective experiments with weights and additional objective values are as specified in subsection 3.1. Due to lower pricing and additional objectives favoring the on-premises operations, we received the expected results, which favor the on-premises placement for all systems, including SID 4 placement due to the network bandwidth costs. From this, we can directly infer the answers to Q1-Q5.

This result was achieved within 700 generations with the size of a population of 200. The best placement result for both single-objective experiments is found on average at generation 428 within 20 experimental runs, while multi-objective NSGA-III required on average 683 generations to achieve the expected outcome.

The presence or absence of placement co-location constraints made no difference in the single-objective scenario, as the estimated network bandwidth's increased pricing expenses negatively affected the objective function's result. However, the multi-objective scenario without constraints required more generations to achieve the same results as with the presence of constraints.

Changing category weights balance presented in Table 1 to favor experience over price with CO weight reduced to $0,10$ and EX increased to $0,50$ also resulted in an expected final landscape placement outcome that favors on-premises for all services because additional objectives specified in this specific case study favor on-premises placement options.

Furthermore, it is also easy to see from Table 1 that the on-premises configuration is favored by the scoring and the pricing in our example is also cheaper on-premises for most systems. In addition to the aforementioned evaluation runs, we have conducted a series of experiments to see if increasing the pricing for on-premises without changing the scoring would affect the decision-making within realistic price ranges. The desired re-allocation favoring on-cloud placement solutions was achieved by increasing the pricing of the on-premises placement by 41% on average, which indicates a strong influence of the customer requirements in the objective function, and hence the algorithms are not guided strictly by the pricing difference. It is important to note that the systems with high volumes of networking communications were also successfully placed in the cloud within the same locations, therefore avoiding high bandwidth costs between regions or between on-premises and on-cloud placement options.

The second set of evaluations we have conducted on a larger IT landscape consisting of 30 SAP systems. The landscape is more heterogeneous in its composition, with 20% of the system estimated to be significantly cheaper when placed in the cloud over the horizon of planning of five years. The rest of the system pricing was almost identical to the first example, with an average margin of difference of 2%. The additional objective values and weights were taken identically to the previous example. However, the networking is not recorded for this specific landscape, and we relied on the expert estimation of the possible bandwidth and system interconnections based on their nature. Only 6 out of 30 systems were designated as interconnected.

The results of this evaluation were consistent with such obtained from data representing a smaller landscape. However, larger problem instances required a greater computational effort from both of the employed algorithms. The best results were obtained with the population size increased to 320, and the number of generations also scaled up to 1200 generations for GA. NSGA-III was able to find a suitable solution with only a 260 population size and within 950 generations. However, these settings are still well within a realm of tolerable computational effort, indicating the potential of the described approach to scale with the size of the landscape.

6 FUTURE WORK

Further investigation is planned on the addition of more constraint types, specifically with a conflicting nature. A mechanism of resolution for constraints conflicts should be investigated. Furthermore, workload forecasting, based on the existing recorded workload profile before placement, is an interesting direction for predicting required capacities according to possible changes in the business environment. Lastly, incorporation of the bin-packing type of a problem for the on-premises placement within the same placement selection process might have the potential to reduce any potential inaccuracies in the cost estimations.

7 CONCLUSION

In this work, we present a simple-to-explain data-driven approach for processing quantifiable requirements and pricing components for the selection of the most suitable placement for commercial off-the-shelf IT enterprise applications, with a case study based on SAP and sizing performed prior to placement based on the recorded real-world system workload profile. We note that this problem can be formulated in a single and multi-objective way, which allows for the potential use of various optimization algorithms. The validity of the approach is evaluated with the use of evolutionary meta-heuristics and the selected algorithms were able to find a suitable solution while taking the pricing, the requirements, and the considered constraints into account. It's also noted that the use of explicit constraints for the facilitation of the co-location for interconnected services leads to faster discovery of a better suitable placement than simple reliance on the implicit increased costs. The approach discussed in this work is suitable for the variable size of considered IT landscapes. However, it's noted that a multi-objective NSGA-III suffers a noticeably smaller performance degradation on larger problems in comparison to the single-objective GA.

REFERENCES

Back, T. (1996). *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, USA, Oxford.

Bichler, M., Setzer, T., and Speitkamp, B. (2006). *Capacity Planning for Virtualized Servers*. Workshop on Information Technologies and Systems (WITS), Milwaukee, Wisconsin, USA, 2006.

Brogi, A., Corradini, A., and Soldani, J. (2019). Estimating costs of multi-component enterprise applications. *Formal Aspects of Computing*, 31(4):421–451.

Chauhan, N., Agarwal, R., Garg, K., and Choudhury, T. (2020). Redundant iaas cloud selection with consideration of multi criteria decision analysis. *Procedia Computer Science*, 167:1325–1333. International Conference on Computational Intelligence and Data Science.

Deb, K. and Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601.

Helbig, M. and Engelbrecht, A. P. (op. 2013). Analysing the performance of dynamic multi-objective optimisation algorithms. In *IEEE Congress on Evolutionary Computation*, pages 1531–1539, [S. l.]. IEEE.

Hippelainen, L., Oliver, I., and Lal, S. (2017). Towards dependably detecting geolocation of cloud servers. pages 643–656. Springer, Cham.

Horký, V., Libič, P., Marek, L., Steinhauser, A., and Tůma, P. (2015). Utilizing performance unit tests to increase performance awareness. In John, L. K., editor, *Proceedings of the 6th ACMSPEC International Conference on Performance Engineering*, pages 289–300, New York, NY. ACM.

Hyser, C., McKee, B., Gardner, R., and Watson, B. J. (2007). Autonomic virtual machine placement in the data center. In *Hewlett Packard Laboratories, Tech. Rep. HPL-2007-189*, volume 189.

Jammal, M. , Kanso, A., and Shami, A. (2015). High availability-aware optimization digest for applications deployment in cloud. In *2015 IEEE International Conference on Communications (ICC)*, pages 6822–6828.

Kruse, R., Borgelt, C., Braune, C., Mostaghim, S., and and Steinbrecher, M. (2011). *Computational intelligence: a methodological introduction*. Springer.

Marler, R. T. and Arora, J. S. (2010). The weighted sum method for multi-objective optimization: new insights. *Structural and Multidisciplinary Optimization*, 41(6):853–862.

Missbach, M., Staerk, T., Gardiner, C., McCloud, J., Madl, R., Tempes, M., and Anderson, G. (2016). *SAP on the Cloud*. Management for Professionals. Springer Berlin Heidelberg, Berlin, Heidelberg, 2nd ed. 2016 edition.

Müller, H., Kharitonov, A., Nahhas, A., Bosse, S., and Turowski, K. (2022). Addressing it capacity management concerns using machine learning techniques. *SN Computer Science*, 3(1):1–15.

Patel, C. D. and Shah, A. J. (2005). Cost model for planning, development and operation of a data center. In *Hewlett-Packard Laboratories Technical Report*, volume 107, pages 1–36.

Pires, F. L. and Baran, B. (2015). A virtual machine placement taxonomy. In *2015 IEEE/ACM 15th International Symposium on Cluster, Cloud and Grid Computing*, pages 159–168, Los Alamitos, California. Conference Publishing Services, IEEE Computer Society.

Rahimi, M., Jafari Navimipour, N., Hosseinzadeh, M., Moattar, M. H., and Darwesh, A. (2022). Toward the efficient service selection approaches in cloud computing. *Kybernetes*, 51(4):1388–1412.

Sarferaz, S. (2022). Data protection and privacy. In *Compendium on Enterprise Resource Planning*, pages 499–513. Springer, Cham.

Wu, C., Buyya, R., and Ramamohanarao, K. (2019). Cloud pricing models: Taxonomy, survey, and interdisciplinary challenges. 52(6).

Zanakis, S. H., Solomon, A., Wishart, N., and Dublish, S. (1998). Multi-attribute decision making: A simulation comparison of select methods. *European Journal of Operational Research*, 107(3):507–529.