Chapter 1

Introduction

1.1 Problem Statement

Data centers are large-scale computing infrastructures which consume huge amount of energy each year: a typical data center consumes as much energy as 25,000 households [10]. Thus, reducing the energy consumption becomes the major concern of Cloud providers. In addition, data centers and computation powers support the modern Cloud computing industry, software industry and etc. Therefore, reducing the cost of data centers will lead to a reduction of cost of softwares which consequently be beneficial to most people who access the Internet on a daily basis. Among several components that consume energy such as cooling system, physical machines (PMs) (e.g servers), and network devices, PMs accounts for 40% and have a huge improvement space, since they are always in low utilization (e.g on average, from 10% to 50% of required resources) [4, 31]. This low utilization of resource problem can be solved by fine granularity management of Cloud resources (e.g CPUs and RAMs) using a new virtuzliation technology: containers [11,15,34] and a new service model: Container as a Service (CaaS) [29]. CaaS is a mixture of traditional IaaS (Infrastructure as a Service) [24] and PaaS (Platform as a Service); it utilizes both containers and virtual machines (VMs) as the fundamental resource management units. In CaaS, applications that were used to deployed in VMs (e.g in IaaS) are now deployed in containers. Container is an operating system (OS) level of virtualization; multiple containers can run on a VM and share OS. Therefore, server consolidation [36] can be applied in a joint of containers and VMs environment to achieve better energy reduction.

Server consolidation is an important stategy in improving the utilization throughout the Cloud resource management processes as shown in Figure 1.1 including new application allocation [17], periodic optimization [25], overloading and under-loading adjustments [25].

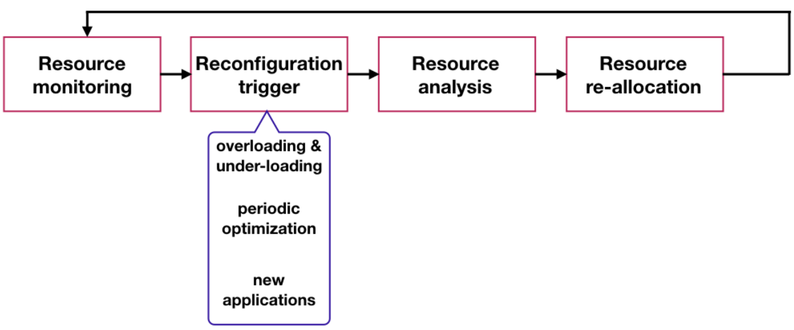

Figure 1.1: A workflow of resource management [25]

According to the characteristic of each process, server consolidation can be roughly classified into two categories: static problems [25] and dynamic problems [6]. Accordingly, server consolidation approaches also have corresponding categories. Static approaches use historical average resource utilization data as input to map applications to PMs. Static consolidation normally involves large amount of applications and PMs, therefore, the optimization is quite time-consuming and often conducted in a off-line fashion. Periodic optimization belongs to this category. It takes a number of existing applications and re-allocate them into a number of PMs. Dynamic approaches take one application each time, allocates it into one of the PMs. The operation is conducted in an online fashion, therefore, it requires fast reaction. Overloading and under-loading can be categories to dynamic consolidation problem [7]. New application allocation can be seen as either static: allocate a batch of new applications, or dynamic: allocate a new application each time. In this proposal, we consider it as a static problem.

Examples of static server consolidation are shown in the following in VM-based and container-based Cloud. In traditional VM-based Cloud, Server consolidation can be described as, given a number of Physical Machines (PMs) which can be represented as the resources(e.g CPU cores and RAM); a number of requests for fixed configurations of VMs (assume applications have been deployed in VMs), each configuration can also be represented as aforementioned resources; The objective is to allocate these requested VMs into a minimum number of PMs. The decision variable is the location of each requested VM. In container-based Cloud, instead of allocating requested VMs in PMs, a set of containers (assume applications have been deployed into containers) represented as resources is first allocated to a number of fixed type VMs, then, these VMs are allocated to PMs. The decision variables are the allocation of containers (upper level) and VMs (lower level). For the upper level of allocation, the objective is to maximize the utilization of resources (e.g a balanced utilization among several resrouces), while the lower level's objective is to minimize the number of PMs.

Traditional VM-based server consolidation are modeled as bin-packing problems [22]. This is because VMs and PMs are naturally modeled as items and bins and server consolidation and bin-packing have the same optimization objective: minimize the number of bins/PMs. The complexity of bin-packing problem is NP-hard which means it is extreme time-consuming to find its optimal solution when the number of decision variables are large. Container-based server consolidation can be categorized as a bilevel optimization problem [9]. Bilevel problems are typically non-convex and strongly NP-hard [38]. In this case, two levels are lower-level: Containers to VM and upper-level: VMs to PMs. These two levels optimization are connected through decision variables. In this case, two levels of optimization are both bin packing problems and they are cooperating [19].

Currently, most research focus on VM-based server consolidation and these methods can not be directly applied on container-based consolidation because of the different structure. Only few research focus on container-based server consolidation problem. One of the state-of-the-art research is from Piraghaj and et al [29]. They first propose a VM-resizing technique that defines the types of VM based on analyzing the historical data from Google. Then they propose a two-step allocation: first allocate containers to VMs and then allocate VMs to PMs. Their major contribution is the method of defining types of VM. The allocation of containers does not optimize the energy consumption and the allocation of VMs are traditional First Fit algorithm. In addition, they propose a dynamic consolidation [28] using a series simple heuristics such as Random Host Selection Algorithm or First Fit Host Selection. Their resource allocation system completely relies on dynamic consolidation without using static methods. Although their system can execute allocation fast, the energy efficiency cannot be guaranteed. The reasons are mainly from two aspects, firstly, they mainly rely on simple bin-packing algorithms to allocate containers to VMs. As Mann's research [22] showed, server consolidation is a lot more harder than bin-packing problem because of the multi-dimensional of resources, many constraints. Therefore, general bin-packing algorithms do not perform well. Secondly, they use a two-step allocation. Because of the interaction of two allocations, separated optimization approach will lead to local optima [23]. Therefore, these two allocations should be considered simultaneously.

The overall goal of this thesis is to develop new container-based server consolidation approaches to solve three problems: joint allocation of containers and VMs, periodic global optimization and dynamic consolidation.

1.2 Motivation

In this thesis, we aim at providing a series of approaches to continuously optimize the joint allocation of VMs and containers. A continuous optimization procedure mainly involves with three types of server consolidation: initialization, global consolidation, and dynamic consolidation. Different stages have distinctive goals, therefore, they are considered as separated research questions. In addition, a scalability problem of static optimization is considered as an optional objective.

1. Joint allocation of containers and VMs (new applications initialization),
   In this research, we take Joint allocation as a static problem which is fundamental for server consolidation problem. At this stage, a set of containers is allocated to a set of VMs and these VMs are allocated to a set of PMs. This task is challenging because the problem is a bilevel optimization where each level is a bin packing problem. Exhaustive search of entire solution space is practically impossible, for the number of possible permutation of solution is huge. Current approaches [16, 28] use simple heuristics such as First Fit to solve the problem. These greedy-based heuristics do not consider the complex structure of the problem, therefore, often reach a local optimal solution.

2. Global consolidation,
   A Global consolidation is conducted to improve the global energy efficiency in a periodical fashion. Data center constantly receives new allocations, releasing of old resources. These changing degrades the compact structure of a data center. Therefore, the data center needs a global optimization to improve the overall energy efficiency.

   The challenges are three folds, firstly, similar with initialization problem, the problem has two level of allocations and they interact with each other. Secondly, like VM-based consolidation, Container-based consolidation is considered as a multi-objective problem with minimization of migration cost as well as keeping a good energy efficiency. In bilevel optimization, multi-objective can be defined in either or both level, therefore, it further increases the complexity. Thirdly, consolidation is a time-dependent process which means the previous solution affects the current decision. Previous VM-based research only consider each consolidation as an independent process. As a consequence, although in one consolidation, the migration is minimized, It may lead to more migrations in the future consolidation. We will consider the robustness of consolidation and propose a novel time-aware server consolidation which takes the previous immediate consolidation and the future consolidation into consideration.

3. Dynamic consolidation,
   It takes one container and allocates it to VMs. Since the size of container can be dynamically adjusted, when the an application is under-provision or over-provision, the

original container is halted, resized and re-allocated. Hence, there is a need to allocate this new container in real time.

To solve a dynamic consolidation, heuristics and dispatching rules are often used [5, 13, 30, 32]. In this scenario, a dispatching rule is considered as a function that determines the priorities of VMs that a container can be placed. However, dynamic placement is much complex than bin-packing problem [22]. Because of its dynamic nature, human designed heuristics are ill-equipped in approximating solutions when the environment has changed [35].

Hyper-heuristic methods, sepcifically, Genetic Programming (GP) technique [3] can learn from the best previous allocation and automatically evolves dispatching rules to solve this problem. GP has been applied in generating dispatching rules for bin-packing problem [8, 35] and other scheduling problems [27]. The results have shown promising results.

There are mainly two challenges, first, it is difficult to identify the related factors that construct the heuristic. Factors or features are the building blocks of heuristics. It is a difficult task because the relationship between a good heuristic and features are not obvious. Second, representations provide different patterns to construct dispatching rules. It is also unclear what representation is the most suitable for the consolidation problem.

4. Large-scale of static server consolidation problem,
   In this case, initialization and global consolidation are belonged to this category. Since Cloud data center typically has hundreds of thousands PMs and more, static server consolidation is always very challenging. Many approaches have been proposed in the literature to resolve the problem. There are mainly two ways, both relied on distributed methods, hierarchical-based [18, 26] and agent-based management systems [41]. The major problem in agent-based systems is that agents rely on heavy communication to maintain a high-level utilization. Therefore, it causes heavy load in the networking. Hierarchical-based approaches are the predominate methods. In essence, these approaches are centralized methods where all the states of PMs within its region are collected and analyzed. The major disadvantage of hierarchical-based approaches is that it only provides local solutions. In fact, it is infeasible and unnecessary to check all the states of PMs since the search space is too large and most PMs do not need a change. This idea motivates a way to improving the effectiveness is to reduce the number of variables so that the search space is narrowed. In this thesis, we are going to investigate the way to eliminate the redundant information.

1.3 Research Goals

1.3.1 Objective One: Develop EC-based approaches for the single objective joint allocation of containers and VMs

Currently, most research focus on VM-based server consolidation technique. They often modeled this problem as a vector bin-packing problem [42]. Container adds an extra layer of abstraction on top of VM. The placement problem has become a two-step procedure, in the first step, containers are packed into VMs and then VMs are consolidated into physical machines. These two steps are inter-related to each other. Previous research [29] solve this problem in separated steps where the first step allocate containers to VMs and the second step allocate VMs to PMs with simple bin-packing heuristics. Therefore, this is the first research that trying to solve the problem.

1. First, our first sub objective is to propose a descriptive single objective model for the bilevel optimization problem of joint allocation of container and VM. The reason to establish this model is because current server consolidation models are mostly VM-based, they cannot be directly applied on bilevel problems. Therefore, variables, constraints and objective functions need to be clarified before applying any optimization algorithm. Each level of the problem will be formulated to a multi-dimensional vector bin packing problem. It is still unclear that which objective function is the best to capture the relationship between container and VM so that the overall energy is low. We will investigate several resource wastage models [12, 14, 40] and select a suitable one. In addition, several models have to be considered, including energy model [10], price model [1], and workload model [21].

2. Second, we will first develop a baseline approach that solve the problem using nested Evolutionary algorithms [33]. We will start from the simplest form: one dimensional bin-packing in each level to more complex multi-dimensional bin-packing.

   Nested methods have been used in solving bilevel problem for years, they are reported as effective approaches. We will investigate several approaches such as Nested Particle Swarm Optimization [20], Differential evolution (DE) based approach [2, 43] and Co-evolutioanry approach [19]. In order to adapt our problem to these existing approaches, we will develop suitable representations and genetic operators.

3. Third, although nested approaches have been reported effective, they are often very time consuming. Therefore, our third sub-objective will focus on developing more efficient algorithms. There are several possible directions to be explored such as metamodeling-based methods [39] and single-level reduction.

1.3.2 Objective Two: Develop EC-based approaches for the multi-objective joint allocation problem

As previous section (see 1.2) mentioned, the task is multi-objective since the number of VM migration has to be minimized while keep the overall energy low. In addition, periodic optimization is a time-dependent problem which means the optimal consolidation in previous operation might lead to more migrations in the current consolidation. The robustness of a data center is particularly important. The robustness measures the stableness of result of consolidation.

1. First, we will develop EC-based approaches to solve the multi-objective joint allocation problem. In this problem, multiple objectives may involve at both of the levels. We will start from a simple case considering multi-objective in lower level: Minimizing VM migration and energy consumption. Currently, there are few the studies using EC methods [**?**, **?**] for multiobjective bilevel optimization. We will investigate which one is more suitable for this binary problem. Furthermore, like the case in single objective problem, we need to develop new representations, genetic operators to apply the algorithms to solve the problem.

2. Second, we will design a robustness measure. Previous studies only use simple measurement which counts the migration number between two static consolidation. This measurement aims at minimizing the number of migration between two static placement processes. It may cause more migration in the next consolidation. Therefore, it needs a time-aware measure of the robustness of system. Therefore, in this objective, the first sub-problem we are going to solve is to propose a robustness measure.

Currently, only a few research propose robustness aware server consolidation techniques [?, ?] have been proposed. They are either static threshold or probability-based threshold to measure the robustness of PMs. We will investigate an adaptive measure based on the historical data and current status.

3. Third, we will design a proactive server consolidation approach. Based on a prediction of future server consolidation and the robustness measure,we will first design an approach which maximize the robustness and also minimize the current energy consumption. Proactive consolidation [?,?] has been studied extensively. Their experience in analyzing the workload pattern can be useful in designing the new algorithm.

1.3.3 Objective Three: Develop a hyper-heuristic Genetic Programming (GP) approach for automatically generating dispatching rules for dynamic consolidation

Previously, dynamic consolidation methods,including both VM-based and container-based, are mostly based on bin-packing algorithm such as First Fit Descending and human designed heuristics. As Mann's research [22] showed, server consolidation is more harder than bin-packing problem because of multi-dimensional of resources and many constraints. Therefore, general bin-packing algorithms do not perform well with many constraints and specific designed heuristics only perform well in very narrow scope. Genetic programming has been used in automatically generating dispatching rules in many areas such as job shop scheduling [27]. GP also has been successfully applied in bin-packing problems [8]. Therefore, we will investigate GP approaches for solving the dynamic consolidation problem. We will start from considering one-level of problem: migrate one VM each time to a PM.

1. First, we will investigate which features and attributes are important when dealing with energy efficiency problem. As the basic component of a dispatching rule, primitive set contains the states of environment including: status of VMs (e.g. utilization, wastage), features of workloads (e.g. resource consumption). Although there is no research has investigate how to use them to construct dispatching rules, there are extensive statistical analysis on workload [37]. The effectiveness of functional set and primitive set will be tested by applying the constructed dispatching rules on dynamic consolidation problem.

2. Develop GP-based methods for evolving Dispatching rules
This sub-objective explores suitable representations for GP to construct useful dispatching rules. It also proposes new genetic operators as well as search mechanisms.

1.3.4 Objective Four (Optional) Large-scale Static Consolidation Problem

Propose a preprocessing method to eliminate redundant variables Current static consolidation takes all servers into consider which will lead to a scalability problem. In this objective, we will propose a method that categorizes servers so that only a small number of servers are considered. This approach will dramatically reduce the search space. The potential approaches that can be applied in this task are various clustering methods.

1.4 Published Papers

During the initial stage of this research, some investigation was carried out on the model of container-based server consolidation [?].

1. Tan, B., Ma, H., Mei, Y. and Zhang, M., "A NSGA-II-based Approach for Web Service Resource Allocation On Cloud". *Proceedings of 2017 IEEE Congress on Evolutioanry Computation (CEC2017).* Donostia, Spain. 5-8 June, 2017.pp.2574-2581

1.5 Organisation of Proposal

The remainder of the proposal is organised as follows: Chapter **??** provides a fundamental definition of the Container-based server consolidation problem and performs a literature review covering a range of works in this field; Chapter **??** discusses the preliminary work carried out to explore the techniques and EC-based techniques for the initialization problem; Chapter **??** presents a plan detailing this projects intended contributions, a project timeline, and a thesis outline.

Bibliography

[1] Al-Roomi, M., and Al-Ebrahim, S. Cloud computing pricing models: a survey. *Computing* (2013).

[2] Angelo, J. S., Krempser, E., and Barbosa, H. J. C. Differential evolution for bilevel programming. In *2013 IEEE Congress on Evolutionary Computation (CEC)* (2013), IEEE, pp. 470–477.

[3] Banzhaf, W., Nordin, P., Keller, R. E., and Francone, F. D. Genetic programming: an introduction, 1998.

[4] Barroso, L. A., and Hölzle, U. The Case for Energy-Proportional Computing. *IEEE Computer 40*, 12 (2007), 33–37.

[5] Beloglazov, A., Abawajy, J. H., and Buyya, R. Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. *Future Generation Comp. Syst. 28*, 5 (2012), 755–768.

[6] Beloglazov, A., and Buyya, R. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers. *Concurrency and Computation - Practice and Experience 24*, 13 (2012), 1397–1420.

[7] Beloglazov, A., and Buyya, R. Managing Overloaded Hosts for Dynamic Consolidation of Virtual Machines in Cloud Data Centers under Quality of Service Constraints. *IEEE Transactions on Parallel and Distributed Systems 24*, 7 (2013), 1366–1379.

[8] Burke, E. K., Hyde, M. R., and Kendall, G. Evolving Bin Packing Heuristics with Genetic Programming. *PPSN 4193*, Chapter 87 (2006), 860–869.

[9] Colson, B., Marcotte, P., and Savard, G. An overview of bilevel optimization. *Annals OR 153*, 1 (2007), 235–256.

[10] Dayarathna, M., Wen, Y., and Fan, R. Data Center Energy Consumption Modeling - A Survey. *... Surveys & Tutorials* (2016).

[11] Felter, W., Ferreira, A., Rajamony, R., and Rubio, J. An updated performance comparison of virtual machines and Linux containers. *ISPASS* (2015), 171–172.

[12] Ferdaus, M. H., Murshed, M. M., Calheiros, R. N., and Buyya, R. Virtual Machine Consolidation in Cloud Data Centers Using ACO Metaheuristic. *Euro-Par 8632*, Chapter 26 (2014), 306–317.

[13] Forsman, M., Glad, A., Lundberg, L., and Ilie, D. Algorithms for automated live migration of virtual machines. *Journal of Systems and Software 101* (2015), 110–126.

[14] Gao, Y., Guan, H., Qi, Z., Hou, Y., and Liu, L. A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. *J. Comput. Syst. Sci. 79*, 8 (2013), 1230–1242.

[15] He, S., Guo, L., Guo, Y., Wu, C., Ghanem, M., and Han, R. Elastic Application Container: A Lightweight Approach for Cloud Resource Provisioning. In *2012 IEEE 26th International Conference on Advanced Information Networking and Applications (AINA)* (Feb. 2012), IEEE, pp. 15–22.

[16] Hindman, B., Konwinski, A., Zaharia, M., Ghodsi, A., Joseph, A. D., Katz, R. H., Shenker, S., and Stoica, I. Mesos - A Platform for Fine-Grained Resource Sharing in the Data Center. *NSDI* (2011).

[17] Jennings, B., and Stadler, R. Resource Management in Clouds: Survey and Research Challenges. *Journal of Network and Systems Management 23*, 3 (2015), 567–619.

[18] Jung, G., Hiltunen, M. A., Joshi, K. R., Schlichting, R. D., and Pu, C. Mistral - Dynamically Managing Power, Performance, and Adaptation Cost in Cloud Infrastructures. *ICDCS* (2010), 62–73.

[19] Legillon, F., Liefooghe, A., and Talbi, E.-G. CoBRA - A cooperative coevolutionary algorithm for bi-level optimization. *IEEE Congress on Evolutionary Computation* (2012), 1–8.

[20] Li, X., Tian, P., and Min, X. A Hierarchical Particle Swarm Optimization for Solving Bilevel Programming Problems. In *Service-Oriented and Cloud Computing*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 1169–1178.

[21] Magalhães, D., Calheiros, R. N., Buyya, R., and Gomes, D. G. Workload modeling for resource usage analysis and simulation in cloud computing. *Computers & Electrical Engineering 47* (2015), 69–81.

[22] Mann, Z. Á. Approximability of virtual machine allocation: much harder than bin packing.

[23] Mann, Z. Á. Interplay of Virtual Machine Selection and Virtual Machine Placement. In *Service-Oriented and Cloud Computing*. Springer International Publishing, Cham, Aug. 2016, pp. 137–151.

[24] Mell, P. M., and Grance, T. The NIST definition of cloud computing. Tech. rep., National Institute of Standards and Technology, Gaithersburg, MD, Gaithersburg, MD, 2011.

[25] Mishra, M., Das, A., Kulkarni, P., and Sahoo, A. Dynamic resource management using virtual machine migrations. *IEEE Communications ... 50*, 9 (2012), 34–40.

[26] Moens, H., Famaey, J., Latré, S., Dhoedt, B., and De Turck, F. Design and evaluation of a hierarchical application placement algorithm in large scale clouds. *Integrated Network Management* (2011), 137–144.

[27] Nguyen, S., Zhang, M., Johnston, M., and Tan, K. C. Automatic Design of Scheduling Policies for Dynamic Multi-objective Job Shop Scheduling via Cooperative Coevolution Genetic Programming. *IEEE Transactions on Evolutionary Computation 18*, 2 (2014), 193–208.

[28] Piraghaj, S. F., Calheiros, R. N., Chan, J., Dastjerdi, A. V., and Buyya, R. Virtual Machine Customization and Task Mapping Architecture for Efficient Allocation of Cloud Data Center Resources. *Comput. J. 59*, 2 (2016), 208–224.

[29] Piraghaj, S. F., Dastjerdi, A. V., Calheiros, R. N., and Buyya, R. Efficient Virtual Machine Sizing for Hosting Containers as a Service (SERVICES 2015). *SERVICES* (2015).

[30] Sarin, S. C., Varadarajan, A., and Wang, L. A survey of dispatching rules for operational control in wafer fabrication. *Production Planning and ... 22*, 1 (Jan. 2011), 4–24.

[31] Shen, S., van Beek, V., and Iosup, A. Statistical Characterization of Business-Critical Workloads Hosted in Cloud Datacenters. *CCGRID* (2015), 465–474.

[32] Shi, W., and Hong, B. Towards Profitable Virtual Machine Placement in the Data Center. *UCC* (2011), 138–145.

[33] Sinha, A., Malo, P., and Deb, K. A Review on Bilevel Optimization: From Classical to Evolutionary Approaches and Applications. *IEEE Transactions on Evolutionary Computation* (2017), 1–1.

[34] Soltesz, S., Pötzl, H., Fiuczynski, M. E., Bavier, A. C., and Peterson, L. L. Container-based operating system virtualization - a scalable, high-performance alternative to hypervisors. *EuroSys 41*, 3 (2007), 275–287.

[35] Sotelo-Figueroa, M. A., Soberanes, H. J. P., Carpio, J. M., Huacuja, H. J. F., Reyes, L. C., and Soria-Alcaraz, J. A. Evolving Bin Packing Heuristic Using Micro-Differential Evolution with Indirect Representation. *Recent Advances on Hybrid Intelligent Systems 451*, Chapter 28 (2013), 349–359.

[36] Varasteh, A., and Goudarzi, M. Server Consolidation Techniques in Virtualized Data Centers: A Survey. *IEEE Systems Journal* (2015), 1–12.

[37] Verma, A., and Dasgupta, G. Server Workload Analysis for Power Minimization using Consolidation. *USENIX Annual Technical Conference* (2009), 28–28.

[38] Vicente, L., Savard, G., and Júdice, J. Descent approaches for quadratic bilevel programming. *Journal of Optimization Theory and Applications 81*, 2 (May 1994), 379–399.

[39] Wang, G. G., and Shan, S. Review of Metamodeling Techniques in Support of Engineering Design Optimization. *Journal of Mechanical Design 129*, 4 (Apr. 2007), 370–380.

[40] Xu, J., and Fortes, J. A. B. Multi-Objective Virtual Machine Placement in Virtualized Data Center Environments. In *Int'l Conference on Cyber, Physical and Social Computing (CPSCom)* (2010), IEEE, pp. 179–188.

[41] Yazir, Y. O., Matthews, C., Farahbod, R., Neville, S. W., Guitouni, A., Ganti, S., and Coady, Y. Dynamic Resource Allocation in Computing Clouds Using Distributed Multiple Criteria Decision Analysis. *IEEE CLOUD* (2010), 91–98.

[42] Zhang, J., Huang, H., and Wang, X. Resource provision algorithms in cloud computing - A survey. *J. Network and Computer Applications 64* (2016), 23–42.

[43] Zhu, X., Yu, Q., and Wang, X. A Hybrid Differential Evolution Algorithm for Solving Nonlinear Bilevel Programming with Linear Constraints. In *2006 5th IEEE International Conference on Cognitive Informatics* (2006), IEEE, pp. 126–131.