Ann Oper Res (2016) 242:1–17
DOI 10.1007/s10479-016-2203-7

Machine reassignment problem: the ROADEF/EURO challenge 2012

H. Murat Afsar[1] · Christian Artigues[2] ·
Eric Bourreau[3] · Safia Kedad-Sidhoum[4]

Published online: 27 April 2016

Abstract The ROADEF/EURO challenge is a contest jointly organized by the French Operational Research and Decision Aid society (ROADEF) and the European Operational Research society (EURO). The contest appears on a regular basis since 1999 and always concerns an industrial optimization problem proposed by an industrial partner. Google proposed a subject for the ROADEF/EURO challenge 2012 (http://challenge.roadef.org/2012/en/), presenting a complex and large-scale machine reassignment problem, where a set of processes assigned to a set of machines have to be reassigned (or moved) while balancing machine usage improvement and moving costs, under resource (more precisely CPU, RAM, disk) and operational constraints. The 2012 challenge edition has been an unprecedented success with 82 registered teams, 48 teams that actually sent a program for qualification, 30 qualified teams and 27 teams that sent a program for the final evaluation. This paper aims at introducing the Annals of Operations Research special issue by presenting the ROADEF/EURO challenge 2012 subject, as well as the methods of the finalist teams and their results.

Keywords ROADEF/EURO challenge · Operations research industrial challenge · Machine reassignment · Cloud computing

1 Introduction

Dillon et al. (2010), meeting the service level agreements is a major issue for cloud computing providers and “effective decision models and optimization algorithms” for efficient resource management are key factors to reach this goal. In this context, Google proposed a

✉ Christian Artigues
artigues@laas.fr

[1] ICD-LOSI, University of Technologie of Troyes, 12, rue Marie Curie, 10004 Troyes, France

[2] LAAS-CNRS, Université de Toulouse, CNRS, Toulouse, France

[3] LIRMM, Université Montpellier 2, 161 rue Ada, 34392 Montpellier Cedex 5, France

[4] UPMC Univ Paris 06, UMR 7606, LIP6, Sorbonne Universités, 75005 Paris, France

subject for the ROADEF/EURO challenge 2012 (http://challenge.roadef.org/2012/en/), presenting a complex and large-scale machine reassignment problem, where a set of processes assigned to a set of machines have to be reassigned (or moved) while balancing machine usage improvement and moving costs, under resource (more precisely CPU, RAM, disk) and operational constraints.

The ROADEF/EURO challenge is a contest jointly organized by the French Operational Research and Decision Aid society (ROADEF) and the European Operational Research society (EURO). The contest appears on a regular basis since 1999 and always concerns an industrial optimization problem proposed by an industrial partner. For the 2012 edition, registered participant teams had to send a first solution program for the qualification phase that ended in December 2011. The programs were evaluated in a first set of training problem instances, the "A" set. The best teams were then qualified for the subsequent stage and could adjust their program to solve a more realistic set of instances that was made available to them, the "B" set. They had to send the revised version of their program in June 2012. The programs were evaluated both on the B set and on a set that was left unknown to the participant, the "X" set. Prizes were awarded at the EURO 2012 conference in Vinius.

In terms of participation, the 2012 challenge edition has been an unprecedented success with 82 registered teams, 48 teams that actually sent a program for qualification, 30 qualified teams and 27 teams that sent a program for the final evaluation. Figure 1 is a map presenting the registered and qualified teams distribution worldwide.

Special issues of journals have been devoted to the challenge since 2005: the car sequencing problem proposed by Renault (Solnon et al. 2008), the workforce scheduling problem proposed by France Telecom (Artigues et al. 2009), the disruption management problem for commercial airlines proposed by Amadeus (Artigues et al. 2012) and the large-scale electricity production problem proposed by EDF (Özcan et al. 2013). This paper aims at introducing this special issue by presenting the ROADEF/EURO challenge 2012 subject, as well as the methods of the finalist teams and their results. Section 2 presents the problem, as defined

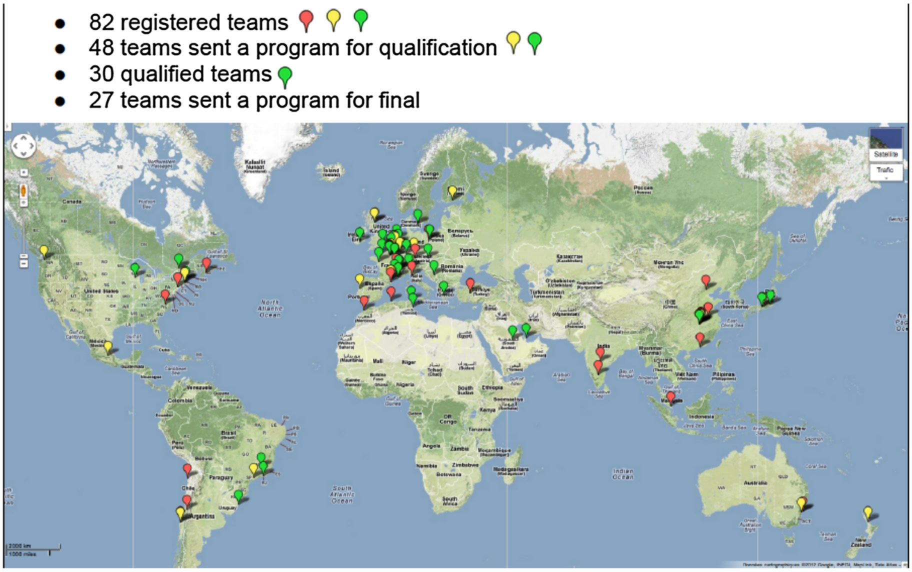
• 82 registered teams
• 48 teams sent a program for qualification
• 30 qualified teams
• 27 teams sent a program for final

Fig. 1 Teams distribution map

by Google. Related work in the literature is described in Sect. 3. The characteristics of the problem instances provided by Google are detailed in Sect. 4. The results of the participant teams for the qualification and for the final stages are analysed in Sect. 5.

2 Problem description

The aim of the addressed problem is to improve the usage of a set of machines related to several resources, such as *RAM* and *CPU*, and processes consuming these resources. Initially each process is assigned to a machine. In order to improve the machine usage, processes can be moved from one machine to another. A solution to this problem is a new process-machine assignment which satisfies all hard constraints and minimizes a given overall cost.
Let $\mathcal{M}$ be the set of machines and $\mathcal{P}$ the set of processes.

Definition 1 (*Solution*) A solution is an assignment of each process $p \in \mathcal{P}$ to one and only one machine $m \in \mathcal{M}$; this assignment is denoted by the mapping $M(p) = m$. The original assignment of process p is denoted $M_0(p)$.

2.1 Constraints

A valid solution is composed of possible moves subject to a set of hard constraints, capacity, spread, dependency and transient constraints.
Let $\mathcal{R}$ be the set of resources which is common to all the machines, $C(m, r)$ be the capacity of resource $r \in \mathcal{R}$ for machine $m \in \mathcal{M}$ and $R(p, r)$ the requirement of resource $r \in \mathcal{R}$ for process $p \in \mathcal{P}$.

Definition 2 (*Usage*) Given an assignment M, the usage U of a machine m for a resource r is defined as:

$$U(m, r) = \sum_{p \in \mathcal{P},\ M(p)=m} R(p, r)$$

Constraint 1 (Capacity) *A process can run on a machine if and only if the machine has enough available capacity on every resource. A feasible assignment must satisfy:*

$$\forall\, m \in \mathcal{M}, r \in \mathcal{R}, \quad U(m, r) \leq C(m, r)$$

Let $\mathcal{S}$ be a set of services which partition the processes.

Constraint 2 (Conflict) *A service $s \in \mathcal{S}$ is a set of processes which must run on distinct machines, i.e.*

$$\forall\, s \in \mathcal{S}, \quad (p_i, p_j) \in s^2, \quad p_i \neq p_j \Rightarrow M(p_i) \neq M(p_j)$$

Let $\mathcal{L}$ be the set of locations, a location $l \in \mathcal{L}$ being a set of machines.

Constraint 3 (Spread) *For each $s \in \mathcal{S}$, let $spreadMin(s) \in \mathbb{N}$ be the minimum number of distinct locations where at least one process of service s should run:*

$$\forall s \in \mathcal{S}, \sum_{l \in \mathcal{L}} min\left(1, \left|\{p \in s \mid M(p) \in l\}\right|\right) \geq spreadMin(s)$$

Let $\mathcal{N}$ be the set of neighborhoods, a neighborhood $n \in \mathcal{N}$ being a set of machines.

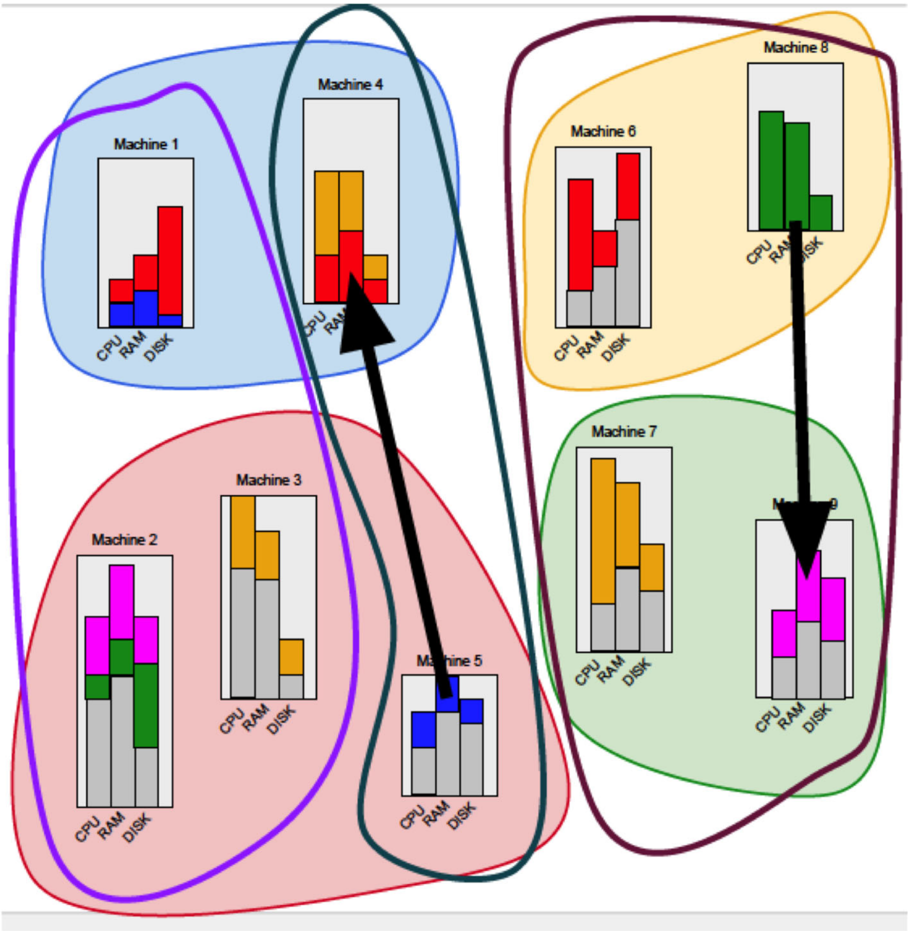

Fig. 2 A small problem instance

Constraint 4 (Dependency) *If service s^a depends on service s^b, then each process of s^a should run in the neighborhood of a s^b process:*

$$\forall\ p^a \in s^a,\ \exists\ p^b \in s^b \text{ and } n \in \mathcal{N} \text{ such that } M(p^a) \in n \text{ and } M(p^b) \in n$$

Let $\mathcal{TR} \subseteq \mathcal{R}$ be the subset of resources which need transient usage.

Constraint 5 (Transient) *When a process p is moved from one machine m to another machine m' some resources are consumed twice; i.e. require capacity on both original assignment $M_0(p)$ and current assignment $M(p)$.*

$$\forall m \in \mathcal{M}, r \in \mathcal{TR}, \quad \sum_{p \in \mathcal{P},\ M_0(p)=m\ \vee\ M(p)=m} R(p,r) \le C(m,r)$$

Figure 2 displays a small problem instance with nine machines on which processes are assigned and occupy a certain amount of resources (CPU, RAM and disk). Each process is a colored rectangle and the color identifies the process service. Machines are doubly partitioned in locations (there are four locations) and neighborhoods (there are three neighborhoods). There are also two dependencies.

2.2 Objectives

The aim of the problem is to improve the usage of a set of machines. To do so, a total objective cost is built by combining a load cost, a balance cost and several move costs.
Let $SC(m, r)$ be the safety capacity of a resource $r \in \mathcal{R}$ on a machine $m \in \mathcal{M}$.

Cost 1 *(Load) The load cost is defined per resource and corresponds to the used capacity above the safety capacity; more formally:*

$$loadCost(r) = \sum_{m \in \mathcal{M}} max\big(0, U(m, r) - SC(m, r)\big)$$

Let $\mathcal{B}$ be a set of triples defined in $\mathcal{R}^2 \times \mathbb{N}$.

Cost 2 *(Balance) For a given triple $b = \langle r_1, r_2, target\rangle \in \mathcal{B}$,*

$$balanceCost(b) = \sum_{m \in \mathcal{M}} max\big(0, target \times A(m, r_1) - A(m, r_2)\big)$$

with $A(m, r) = C(m, r) - U(m, r)$.

Let $PMC(p)$ be the cost of moving the process p from its original machine $M_0(p)$.

Cost 3 *(Process Move)*

$$processMoveCost = \sum_{\substack{p \in \mathcal{P} \text{ such that} \\ M(p) \neq M_o(p)}} PMC(p)$$

Cost 4 *(Service Move)*

$$serviceMoveCost = \max_{s \in \mathcal{S}} \Big(\big|\{p \in s \mid M(p) \neq M_0(p)\}\big| \Big)$$

Let $MMC(m_{source}, m_{destination})$ be the cost of moving any process p from machine m_{source} to machine $m_{destination}$.

Cost 5 *(Machine move)*

$$machineMoveCost = \sum_{p \in \mathcal{P}} MMC(M_0(p), M(p))$$

2.2.1 Total objective cost

The objective is to minimize the weighted sum of all the previous cost components.

$$\begin{aligned} totalCost = & \sum_{r \in \mathcal{R}} weight_{loadCost}(r) \cdot loadCost(r) \\ & + \sum_{b \in \mathcal{B}} weight_{balanceCost}(b) \cdot balanceCost(b) \\ & + weight_{processMoveCost} \cdot processMoveCost \\ & + weight_{serviceMoveCost} \cdot serviceMoveCost \\ & + weight_{machineMoveCost} \cdot machineMoveCost \end{aligned}$$

3 Related work

Cloud computing aims at enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction (Dillon et al. 2010). The resource managers need to employ fast and effective decision models and optimization algorithms. The topic proposed for the ROADEF/EURO challenge aims at providing such algorithms. Some works have been conducted in this area. We can quote the following works.

Xu et al. (2014) review the state-of-the-art research on managing the performance overhead of virtual machines, and summarize them under diverse scenarios of the Infrastructure-as-a-Service (IaaS) cloud, ranging from the single-server virtualization, a single mega datacenter, to multiple geodistributed datacenters. Specifically, they discuss the performance modeling methods with a particular focus on their accuracy and cost, and compare the overhead mitigation techniques by identifying their effectiveness and implementation complexity. Yang and Tate (2012) present a descriptive literature review and classification scheme for cloud computing research that consists of four main categories: technological issues, business issues, domains and applications, and conceptualising cloud computing. We can also mention the review of Zhao et al. (2014) on the main challenges of the cloud computing paradigm.

Setzer and Stage (2010) propose a method that aims at the reduction of managerial complexity of resource and workload management in data centers hosting thousands of applications with varying workload behaviors. Their method is based on determining points in time where migrations are likely to be beneficial for a given set of workloads.

Lin et al. (2011) consider that power consumption is one of the most critical problems in data centers. One effective way to reduce power consumption is to consolidate the hosting workloads and shut down physical machines which become idle after consolidation. They show that server consolidation is an NP-hard problem. They propose a dynamic round-robin algorithm, for energy-aware virtual machine scheduling and consolidation. Cambazard et al. (2013) address the allocation of virtual machines to servers with time-variable resource demands in data centers in order to minimize energy costs while ensuring service quality. They present a scalable constraint programming-based large neighborhood search method. Kessaci et al. (2013) propose a multi-objective genetic algorithm to optimize the energy consumption, carbon dioxide emissions and the generated profit of a geographically distributed cloud computing infrastructure. Wang et al. (2014) propose a multi-objective bi-level programming model to improve the energy efficiency of servers. They combine an energy-aware data placement policy and a locality-aware multi-job scheduling scheme. Chang et al. (2010) formulate demand for computing power and other resources as a resource allocation problem with multiplicity, where computations that have to be performed concurrently are represented as tasks and a later task can reuse resources released by an earlier task. They show that finding a minimal allocation is NP-complete and present an approximation algorithm. Mezmaz et al. (2011) investigate the problem of scheduling precedence-constrained parallel applications on heterogeneous computing systems like cloud computing infrastructures. They propose a parallel bi-objective hybrid genetic algorithm that takes into account, not only makespan, but also energy consumption.

In this general context, the problem considered in the ROADEF/EURO challenge aims at considering, at a single step, the dynamics of the virtual machine assignment process and, more precisely, the benefit of machine reassignment.

4 Instances

Tables 1 and 2 present the characteristics of data sets A, B and X. Data set A was used to rank the participants during the qualification phase. Data sets B and X were used to rank the participants for the final stage. Data set X has been kept unknown to the participants throughout the competition. In dataset A, the number of machines range from 4 to 100 and the number of processes range from 100 to 1000. In larger datasets B and X the number of machines ranges from 100 to 5000 and the number of processes ranges from 5000 to 50,000. There are ten instances per set. Note that each instance of the dataset X has exactly the same parameter range that its corresponding B instance.

Table 1 Characteristics of dataset A

	a1_1	a1_2	a1_3	a1_4	a1_5	a2_1	a2_2	a2_3	a2_4	a2_5
#Machines	4	100	100	50	12	100	100	100	50	50
#Resources	2	4	3	3	4	3	12	12	12	12
#Transient	0	1	1	1	1	0	4	4	0	0
#Services (non unit)	10	10	100	100	10	0	100	125	125	125
#Neighborhoods	1	2	5	50	2	1	5	5	5	5
#Locations	4	4	25	50	4	1	25	25	25	25
#Dependencies (max per service)	10	10	10	10	10	0	0	10	10	10
#Balance	1	0	0	1	1	0	0	0	1	0

Table 2 Characteristics of datasets B and X

	1	2	3	4	5	6	7	8	9	10
#Machines	100	100	100	500	100	200	4000	100	1000	5000
#Resources	12	12	6	6	6	6	6	3	3	3
#Transient	4	0	2	0	2	0	0	1	0	0
#Services (non unit)	500	500	1000	1000	1000	1000	1000	1000	1000	1000
#Neighborhoods	5	5	5	5	5	5	5	5	5	5
#Locations	10	10	10	50	10	50	50	10	100	100
#Dependencies (max per service)	30	30	50	60	60	60	50	60	60	70
#Balance	0	1	0	1	0	1	1	0	1	1
#Average slack (%)	15	15	20	20	20	20	30	20	10	20
Turn down (%)	4	0	4	0	4	0	0.5	4	0	0.4

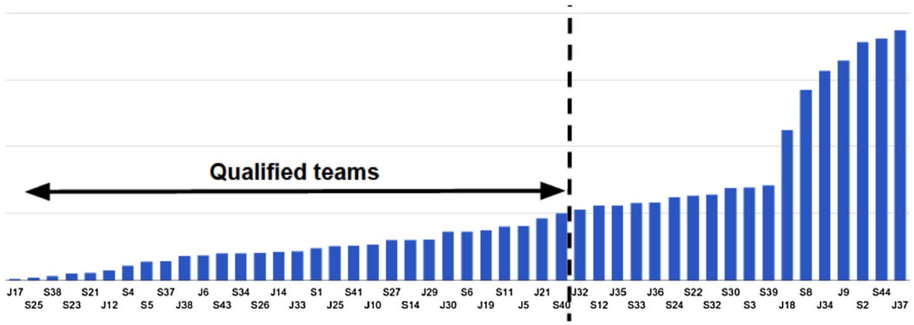

Fig. 3 Rank of the teams for the qualification stage

5 Results and methods overview

5.1 Results

A trivial solution corresponds to the solution where no process is reassigned. We call this solution the reference solution. For both the qualification and the final stages, teams were ranked according to the same scheme. The score of a team for a given instance is given by the gap of the team's cost from the best cost divided by the cost of the reference solution. For the qualification stage, the score of a team was the sum of its scores on all instances of dataset A. For the final stage, the score of a team was the sum of its scores on all B and X dataset instances.

For the qualification stage a threshold for the qualification has been set to a score of 100 %. Figure 3 shows a chart displaying the score of each competing team and the qualified teams. Team identifier either starts with an "S" (resp. "J") for Senior (resp. Junior) team. The condition for a team to be allowed to compete in the junior category is that it is made only by students, without any completed PhD. Any other team is registered in the Senior category. For more information on the registered teams, we refer to the 2012 challenge web site (http://challenge.roadef.org/2012/en/). We give a more detailed description of the qualified team composition and results for the qualification stage in Table 3. It is worth noticing that the best team in the qualification stage was the junior team J17 (Wauter and Vancroonenburg).

For the final stage, the qualified participants had to deal with the much larger instances (see Table 2). Among the qualified teams, 27 teams sent a program for evaluation. The results of these 27 teams are summarized in Fig. 4 and in Table 4.

To evaluate the instance difficulty, with regard to the results of the participant methods, Fig. 5 displays the percentiles among the participants of the score for each instance of the B and X datasets. The figure shows that some instances were highly discriminating (e.g. B2, X7).

5.2 Methods overview

Below we provide a brief description of the method of each participant team which clearly illustrates the diversity of the proposed optimization approaches. We only mention quantitative results for the teams that obtained the Best Known Solutions (BKS) on at least one

Table 3 Results of the qualified teams for the qualification phase

Rank	Team id	Team members	Score
1	J17	T. Wauter, W. Vancroonenburg (Belgium)	1.73
2	S25	A. Gharbi (Saudi Arabia) M. Haouari (Tunisia), M. Mrab (Saudi Arabia) M. Kharbeche (Qatar)	3.58
3	S38	D. Mehta, B. O'sulluvan, H. Simonis (Ireland)	5.92
4	S23	M. Ritt, L.S. Buriol, G. Portal, L. Borba, A.J. Benavides (Brazil)	9.55
5	S21	S. Hanafi, M. Vasquez, Y. Vimont (France), M. Yagiura, H. Haschimoto, K. Nonobe (Japan)	10.97
6	J12	W. Jaśkowski, P. Gawron, M. Szubert, B. Wieloch (Poland)	14.58
7	S4	A. Bloemen (Netherlands)	22.06
8	S5	C. Gogos, C. Valouxis, P. Alefragis, E. Housos (Greece)	27.98
9	S37	Z. Lu, T. Ye, Z. Whang (China)	28.37
10	J38	N. Catusse (France)	36.64
11	J6	H. Dudebout, R. Masson, J. Michallet (France), P.H.V. Petrucci, A. Subramanian (Brazil), T. Vidal (France)	37.32
12	S43	D. Chemla, B. Gacias, P. Gianessi (France)	40.49
13	S34	M. Pécot (France)	40.61
14	S26	L. Alfandari, F. Butelle, C. Coti, L. Finta, G. Plateau, F. Roupin, A. Rozencnop (France)	41.30
15	J14	M. Ruiz (France)	42.44
16	J33	A. Sansoterra, L. Ferruci, N. Calcaveccia, F. Sironi (Italy)	43.79
17	S1	J. Peekstok, E. Kuipers (Netherlands)	47.35
18	J25	F. Brandt, M. Völker, J. Speck (Germany)	50.59
19	S41	M. Buljabašić, E. Demirović, H. Gavranović (Bosnia)	51.15
20	J10	M. Larose, M. Posta (Canada)	52.90
21	S27	C. de Oliveira,R.P. Lopes, T.F. de Noronha, V.W. de Morais, V.A.A. de Souza (Brazil)	59.81
22	S14	N. Teypaz (France)	59.82
23	J29	J. Holzer, L. Tang, Y. Liu, A. Gore, L. Watkins (USA)	60.74
24	J30	S. Pirkwieser, R. Steiner (Austria)	72.17
25	S6	T. Benoist, B. Estellon, F. Gardi, R. Megel, J. Darlay, K. Nouioua (France)	72.43
26	J19	S. Zaourar, M. Gabay (France)	74.60
27	S11	F. Clautiaux, A. Liefooghe, F. Legillon, E.-G. Talbi (France)	80.11
28	J5	K. Tierney, A. Delgado, D. Pacino, Y. Malitsky (Denmark)	80.77
29	J21	V. Chiraphadhanakul, C. Figueroa (USA)	92.60
30	S40	B. Jarbou (Tunisia), N. Mladenovic (United Kingdom)	99.63

instance, or a gap close to the best known results on some instances. We also underline that, due to the exceptional participation rate, being in this short list was a real challenge. Table 5 gives a short and rought overview of the components of each medhod where the columns refer to the following components: LB (a Lower Bound method was inculded), LS (Local search, other than LNS, HH, GRASP, TS, SA and VNS was used), LNS (Large Beigh-

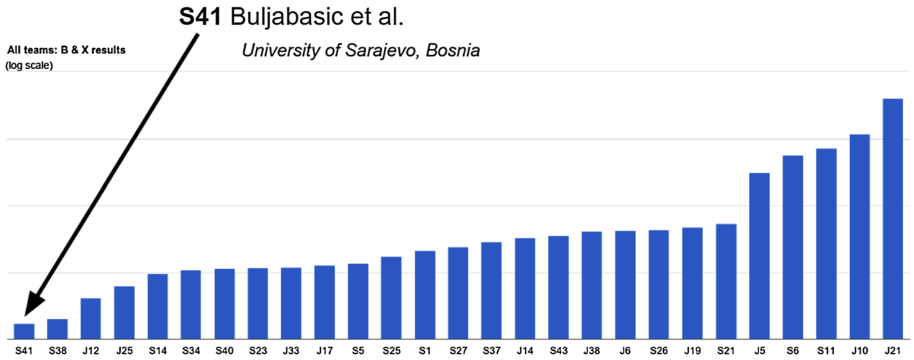

Fig. 4 Final ranking

borhood Search), HH (Hyper-Heuristic), GRASP (Greedy Randomized Adaptive Search Procedure), TS (Tabu Search), SA (Simulated Annealing), VNS (Variable Neighborhood Search), LA (Late Acceptance heuristic), PR (Path-Relinking), CP (Constraint Programming), (MI)LP (Mixed-Integer Linear Programming or Linear Programming was used), DP (Dynamic Programming), PS (a Parallel Search component was a key element of the method).

Buljabašić, Demirović and Gavranović, (winning team S41, paper in this special issue) propose two lower bounds based on load and balance costs to evaluate the quality of their local search algorithm. The method exploits four neighbourhoods: shift, swap, Big Process Rearrangement (BPR) and chain shift. They claim that BPR influences the objective function much more than the small process rearrangement. To choose a good set of processes, they construct an auxiliary directed weighted graph where each node represents a process. They find two BKS in dataset X and three BKS in dataset B.

Mehta, O'Sullivan and Simonis (Team S38) (Mehta et al. 2012; Malitsky et al. 2013) present a Constraint Programming (CP) model with large neighbourhood search. At each iteration of the algorithm, a subset of processes to be reassigned is selected and the variable domains of the CP model is updated. The resulting CP is solved with a threshold on the number of failures and the best solution is kept for the following iteration. They also propose lower bounds summing load and balance costs and report 0.26 % of gap. They find two BKS in dataset X and one BKS in dataset B.

The method of Jaśkowski, Gawron, Szubert and Wieloch (Team J12, paper in this special issue) consists in combining a mixed integer linear programming and a heuristic approach. The method has three phases: a greedy hill climber to obtain a good starting solution with moves on processes and machines, a hyper-heuristic approach based on several low-level heuristics and finally a Mixed Integer linear Programming (MIP) with randomized moves. The MIP heuristic iteratively selects a small subset of machines and reassigns optimally the processes to the subset of machines using a MIP solver. The choice of the machine subset can be randomized or decided by a dynamic programming. This method finds one BKS in dataset X.

Another successful combination of a large neighbourhood search with a constraint programming model is proposed by Brandt, Völker and Speck (Team J25, paper in this special issue). Their hybrid method uses multiple threads. Iteratively, a subset of processes is chosen by neighbourhood search and CP model reassigns these processes. At the end of each iteration, the best solution of each thread is synchronized. On seven instances over ten of the X

Table 4 Final scores

Rank	Team id	Team members	Score
1	S41	M. Buljabašić, E. Demirović, H. Gavranović (Bosnia)	0.47
2	S38	D. Mehta, B. O'sulluvan, H. Simonis (Ireland)	0.62
3	J12	W. Jaśkowski, P. Gawron, M. Szubert, B. Wieloch (Poland)	1.72
4	J25	F. Brandt, M. Völker, J. Speck (Germany)	2.60
5	S14	N. Teypaz (France)	3.91
6	S34	M. Pécot (France)	4.32
7	S40	B. Jarbou (Tunisia), N. Mladenovic (UK)	4.56
8	S23	M. Ritt, L.S. Buriol, G. Portal, L. Borba, A.J. Benavides (Brazil)	4.58
9	J33	A. Sansoterra, L. Ferruci, N. Calcaveccia, F. Sironi (Italy)	4.66
10	J17	T. Wauter, W. Vancroonenburg (Belgium)	4.95
11	S5	C. Gogos, C. Valouxis, P. Alefragis, E. Housos (Greece)	5.28
12	S25	A. Gharbi (Saudi Arabia) M. Haouari (Tunisia), M. Mrab (Saudi Arabia) M. Kharbeche (Qatar)	6.41
13	S1	J. Peekstok, E. Kuipers (Netherlands)	7.49
14	S27	C. de Oliveira,R.P. Lopes, T.F. de Noronha, V.W. de Morais, V.A.A. de Souza (Brazil)	8.34
15	S37	Z. Lu, T. Ye, Z. Whang (China)	9.60
16	J14	M. Ruiz (France)	10.66
17	S43	D. Chemla, B. Gacias, P. Gianessi (France)	11.23
18	J38	N. Catusse (France)	12.53
19	J6	H. Dudebout, R. Masson, J. Michallet (France), P.H.V. Petrucci, A. Subramanian (Brazil), T. Vidal (France)	12.90
20	S26	L. Alfandari, F. Butelle, C. Coti, L. Finta, G. Plateau, F. Roupin, A. Rozencnop (France)	13.16
21	J19	S. Zaourar, M. Gabay (France)	14.07
22	S21	S. Hanafi, M. Vasquez, Y. Vimont (France), M. Yagiura, H. Haschimoto, K. Nonobe (Japan)	15.39
23	J5	K. Tierney, A. Delgado, D. Pacino, Y. Malitsky (Denmark)	54.46
24	S6	T. Benoist, B. Estellon, F. Gardi, R. Megel, J. Darlay, K. Nouioua (France)	83.91
25	S11	F. Clautiaux, A. Liefooghe, F. Legillon, E.-G. Talbi (France)	99.16
26	J10	M. Larose, M. Posta (Canada)	140.05
27	J21	V. Chiraphadhanakul, C. Figueroa (USA)	336.19

dataset, they have less than 1.6 % of gap from the BKS. However, on some instances like X3 and X5, they are very far from the best known solution, 1883.36 and 324.66 % respectively.

Teypaz (Team S14) also proposes an integration of an exact algorithm with a metaheuristic. The algorithm is a tabu search with two phases using matching moves. To evaluate these moves, a maximum weight matching problem is solved by the well-known blossom shrinking algorithm of Edmonds. The tabu search uses two elementary moves: a reassignment of process and a swap of two processes. This algorithm is provided by the open source LEMON library and finds one BKS in dataset X.

Another tabu search is proposed by Pécot (Team S34) with two neighbourhoods: insert and switch. Instead of the whole neighbourhoods, only randomly picked subsets are used.

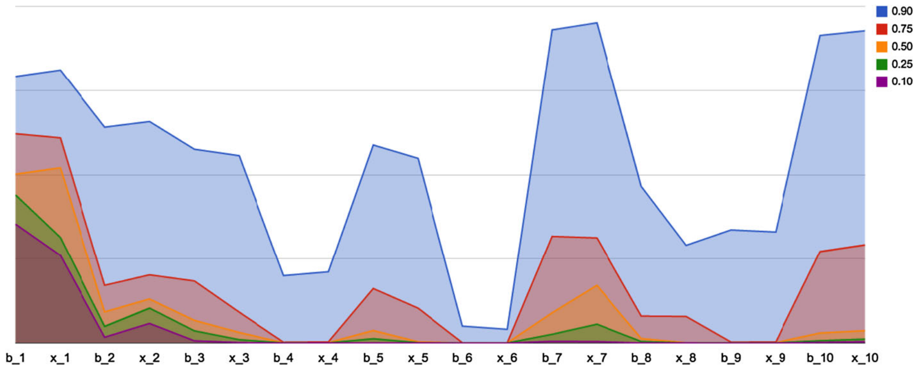

Fig. 5 Percentiles for the final results

This simple method is very efficient and finds two BKS in the X instances and three BKS in the B instances.

Jarbou and Mladenovic (Team S40) presents a three level Variable Neighborhood Search (VNS) algorithm which decomposes problem into smaller size problems. The main structure, skewed variable neighborhood search, may accept slightly worse solutions, if they are different enough from the incumbent solution. This method finds two BKS on the B instances.

The Simulated Annealing (SA) of Ritt, Buriol, Portal, Borba and Benavides (Team S23, paper in this special issue) has two neighbourhoods (process reassignment and process swap) which are selected randomly. They propose a particular data structure which allows to evaluate moves and perform updates in constant time. On six instances of the X dataset, they have less than 0.2 % of gap from the BKS. Moreover, the maximum gap from the BKS is less than 50 % (48.7 % on instance X5).

Sansoterra, Ferruci, Calcaveccia and Sironi (Team J33) propose a parallel simulated annealing and variable neighbourhood search, which run concurrently to exploit multiple CPU cores. Communication between heuristics happens through a shared solution pool. This solution pool provides a safe access to two separate solution sets, one with high quality and the other with high diversity solutions. A path relinking procedure takes one solution from each set with a probability of 50 % and explores intermediate solutions. One BKS from dataset X and one BKS from dataset B are found.

A late acceptance hill climbing metaheuristic with two simple and fast neighbourhood functions is proposed by Vancroonenburg and Wauters (Team J17). This methods accepts a new solution if it is better than the current solution of the last L iterations. As some of the former teams, the two moves are process reassignment and swap machine neighbourhood. Preliminary tests show that a greater value of L leads to a slower convergence but to a better final solution. On six instances of set X, the maximum gap from the BKS is less than 1.5 % but on some instances such as X5, the gap can be very large.

A hybrid method combining integer programming and a late acceptance metaheuristic with large variable neighbourhood search is proposed by Gogos, Valouxis, Alefragis and Housos (Team S5). These components are cooperatively executed until the time limit. Even if the team S5 finds no BKS in either sets, the maximum gap from the BKS is less than 234.94 % in the worst case.

Team S25 (Gharbi, Haouari, Mrab and Kharbeche, in this special issue) proposes a mixed integer linear model for the problem. However, because of the size of the instances, the MIP is iteratively solved on small subsets of machines and assigned processes. The maximum gap

Table 5 Overview of used techniques

Team	Method components													
	LB	LS	LNS	HH	GRASP	TS	SA	VNS	LA	PR	CP	(MI)LP	DP	PS
M. Buljabašić, E. Demirović, H. Gavranović (Bosnia)	✓	✓												
D. Mehta, B. O'sulluvan, H. Simonis (Ireland)	✓		✓								✓			
W. Jaśkowski, P. Gawron, M. Szubert, B. Wieloch (Poland)		✓		✓								✓	✓	
F. Brandt, M. Völker, J. Speck (Germany)			✓								✓			✓
N. Teypaz (France)						✓								
M. Pécot (France)						✓								
B. Jarbou (Tunisia), N. Mladenovic (UK)								✓						
M. Ritt, L.S. Buriol, G. Portal, L. Borba, A.J. Benavides (Brazil)							✓							
A. Sansoterra, L. Ferruci, N. Calcaveccia, F. Sironi (Italy)							✓	✓		✓				✓
T. Wauter, W. Vancroonenburg (Belgium)		✓							✓					
C. Gogos, C. Valouxis, P. Alefragis, E. Housos (Greece)			✓					✓	✓			✓		
A. Gharbi (Saudi Arabia) M. Haouari (Tunisia), M. Mrab (Saudi Arabia) M. Kharbeche (Qatar)			✓									✓		
J. Peekstok, E. Kuipers (Netherlands)		✓					✓					✓		
C. de Oliveira,R.P. Lopes, T.F. de Noronha, V.W. de Morais, V.A.A. de Souza (Brazil)		✓						✓				✓		

Table 5 continued

Team	Method components													
	LB	LS	LNS	HH	GRASP	TS	SA	VNS	LA	PR	CP	(MI)LP	DP	PS
Z. Lu, T. Ye, Z. Whang (China)		✓												
M. Ruiz (France)								✓				✓		
D. Chemla, B. Gacias, P. Gianessi (France)		✓										✓		
N. Catusse (France)		✓												
H. Dudebout, R. Masson, J. Michallet (France), P.H.V. Petrucci, A. Subramanian (Brazil), T. Vidal (France)		✓	✓									✓		
L. Alfandari, F. Butelle, C. Coti, L. Finta, G. Plateau, F. Roupin, A. Rozencnop (France)			✓				✓							✓
S. Zaourar, M. Gabay (France)					✓									
S. Hanafi, M. Vasquez, Y. Vimont (France), M. Yagiura, H. Haschimoto, K. Nonobe (Japan)		✓				✓						✓		
K. Tierney, A. Delgado, D. Pacino, Y. Malitsky (Denmark)			✓			✓						✓		✓
T. Benoist, B. Estellon, F. Gardi, R. Megel, J. Darlay, K. Nouioua (France)		✓												
F. Clautiaux, A. Liefooghe, F. Legillon, E.-G. Talbi (France)		✓										✓		
M. Larose, M. Posta (Canada)		✓								✓				✓
V. Chiraphadhanakul, C. Figueroa (USA)		✓										✓		

from the BKS on seven instances of the X dataset is less than 3.06 % but the maximum gap is significantly larger on some other instances.

A three phase algorithm combining a simulated annealing, steepest descent and linear model is used by Peekstok and Kuipers (Team S1) Although the initial algorithm took into account the infeasible solutions, with a penalty cost, because of the size of the solution space implied, in the final form of the method only feasible solutions are considered. They obtain one BKS on the instances of dataset X.

Another hybridization of integer programming with a metaheuristic is introduced by de Oliveira, Lopes, de Noronha, de Morais and de Souza (Team S27). Their iterated local search using variable neighbourhood descent has a perturbation procedure based on the IP formulation. They report an average improvement of 63.97 % of the initial solution.

A more classical iterated local search is proposed by the team Lu and Whang (Team S37) exploring three neighbourhoods: process reassignment, 1-1 process swap and 1-2 process swap. When the incumbent solution is not updated, a random number of moves is applied to perturb the actual solution. On seven instances of set X, they have less than 4.83 % of gap from the BKS but the maximum gap on all X instances is very large.

Variable neighborhood descent of the Ruiz (Team J14) includes three descent algorithms which are randomly applied. One of these algorithms is a MIP.

Chemla, Gacias and Gianessi (Team S43) use a local search algorithm that provides a warm start to a MIP. For large instances, an aggregation phase is needed.

Catusse (Team J38) proposes a hill climbing heuristic using three neighborhoods: *Move* changes the assignment of a process from one machine to other, *Swap* exchanges two processes of two machines and *2-1 Swap* exchanges 2 processes against 1. A new solution is accepted if it improves the incumbent solution, but sometimes to accelerate the convergence, a degrading solution may be accepted.

Dudebout, Masson, Michallet, P.H.V. Petrucci, A. Subramanian, T. Vidal (Team J6) give a hybridization of a large neighborhood search and a local search. Local search uses two basic moves: *Relocation* and *swap* of processes on the machines. The large neighborhood search iteratively destroys and reconstructs large part of the solution. For the reconstruction, a mixed integer program is executed.

Alfandari, Butelle, Coti, Finta, Plateau, Roupin and Rozencnop (Team S26, paper in this special issue) run two heuristics, a large neighborhood search and a simulated annealing, each one on a different thread, under a framework aiming at electing the best algorithm depending of the feedback during search, based on ideas close to the hyper-heuristic concept.

Zaourar and Gabay (Team J19, in this special issue) propose a greedy randomized adaptive search procedure for the vector bin packing problem with heterogeneous bin sizes, which uses a first fit decreasing greedy algorithm for the construction phase. They provide results on instances of the vector bin packing and show how the heuristics can be adapted to the challenge problem.

Hanafi, Haschimoto, Nonobe, Vasquez, Vimont and Yagiura (Team S21) combine a mixed integer linear formulation and two metaheuristics: iterative local search and tabu search with reverse elimination method to maintain the tabu list. At each iteration, the incumbent solution is replaced by the best solution in the union of two neighborhoods: *Shift* and *Swap*. A solution is better if an aggregated objective value is less. Each component of the objective function is weighted and the weights are updated when a local optimum is found.

Another parallel approach is developed by Tierney, Delgado, Pacino and Malitsky (Team J5). They propose a large neighborhood search using a mixed integer model in a destruction/reconstruction principal. Randomly selected variables of the incumbent solution are relaxed and MIP is called as a local search operator to find the best values for relaxed variables. The iterations continue until a termination criteria is met.

Benoist, Estellon, Gardi, Megel, Darlay and Nouioua (Team S6) translate the problem into a binary model and run the default search of Local Solver; a commercial solver based on local search which uses an adaptive simulated annealing as search heuristic. A move of the local search consists of flipping the values of a certain number of binary variables and computing the resulting objective function. All moves resulting to the violation of some constraint are rejected. Other moves are accepted depending on their impact on the objective function.

Clautiaux, Liefooghe, Legillon and E.-G. Talbi (Team S11) propose a method divided into two phases, applied one after the other, in a sequential way. In the first stage, an iterated local search algorithm is performed. During the second phase, a heuristic aims at re-balancing the overload cost over machines. They use an integer programming solver to improve locally the incumbent solution.

Larose and Posta (Team J10) run a parallel version of the iterative local search on two threads of the test machine. While iterative local search generates a pool of feasible solutions, they are improved by a path-relinking phase. One iteration of the path relinking consists in randomly selecting an initial and guiding solution from the pool. Then it generates a neighborhood based on moving a process from machine to another.

Chiraphadhanakul and Figueroa (Team J21) decompose the problem into subproblems to balance the load of a subset of machines and to swap processes following two selection methods to minimize the overall costs, repeatedly. Each subproblem starts by solving a linear relaxation of the unconstrained problem to obtain the reduced costs. Then, using the reduced costs, they estimate how costly it is to assign a process on a particular machine.

6 Concluding remarks

As already mentioned the ROADEF/EURO challenge 2012 was a tremendous successful scientific event. We believe that the numerous proposed methods that includes local search, metaheuristics, matheuristics, hybrid constraint-programming and integer programming components made a significant advance of the state of the art for industrial virtual machine reassignment problem.

Further research should concentrate on the dynamic aspects of the problem considering several reassignment steps over a delimited horizon. This questions predictability aspects and makes it necessary to propose intelligent reactive procedures as well as robust optimization methods.

Acknowledgments The authors warmly thank Emmanuel Guere, Google, for his high involvement in the challenge, in particular for the subject definition and the results analysis.

References

Artigues, C., Bourreau, E., Afsar, H.-M., Briant, O., & Boudia, M. (2012). Disruption management for commercial airlines: Methods and results for the ROADEF 2009 challenge. *European Journal of Industrial Engineering*, *6*(6), 669–689.

Artigues, C., Cung, V.-D., Finke, G., & Penz, B. (2009). Editorial articles ROADEF 2007. *RAIRO Operations Research*, *43*(4), 373–374.

Cambazard, H., Mehta, D., O'Sullivan, B., & Simonis, H. (2013). Constraint programming based large neighbourhood search for energy minimisation in data centres. In J. Altmann, K. Vanmechelen & O. F. Rana (Eds.), *Economics of grids, clouds, systems, and services* (pp. 44–59). Berlin: Springer.

Chang, F., Ren, J., & Viswanathan, R. (2010). Optimal resource allocation in clouds. In *2010 IEEE 3rd international conference on cloud computing (CLOUD)* (pp. 418–425). IEEE.

Dillon, T., Wu, C., & Chang, E. (2010). Cloud computing: Issues and challenges. In *24th IEEE international conference on advanced information networking and applications (AINA)* (pp. 27–33).

Kessaci, Y., Melab, N., & Talbi, E.-G. (2013). A Pareto-based genetic algorithm for optimized assignment of virtual machines requests in a multi-cloud environment. In *IEEE conference on evolutionary computation (CEC)*, Cancun, Mexique, June.

Lin, C.-C., Liu, P., & Wu, J-J. (2011). Energy-aware virtual machine dynamic provision and scheduling for cloud computing. In *2011 IEEE international conference on cloud computing (CLOUD)* (pp. 736–737). IEEE.

Malitsky, Y., Mehta, D., O'Sullivan, B., & Simonis, H. (2013). Tuning parameters of large neighborhood search for the machine reassignment problem. In C. Gomes & M. Sellmann (Eds.), *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR)* (pp. 176–192). Berlin: Springer.

Mehta, D., O'Sullivan, B., Simonis, H. (2012). Comparing solution methods for the machine reassignment problem. In M. Milano (Ed.), *Principles and Practice of Constraint Programming* (pp. 782–797). Heidelberg: Springer.

Mezmaz, M., Melab, N., Kessaci, Y., Lee, Y. C., Talbi, E.-G., Zomaya, Albert Y., et al. (2011). A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems. *Journal of Parallel and Distributed Computing*, *71*(11), 1497–1508.

Özcan, E., Artigues, C., Bourreau, E., & Afsar, H.-M. (2013). Special issue on maintenance scheduling: Theory and applications. *Journal of Scheduling*, *16*(6), 549–550.

ROADEF/EURO. Challenge 2012. http://challenge.roadef.org/2012/en/.

Setzer, T., & Stage, A. (2010). Decision support for virtual machine reassignments in enterprise data centers. In *Network operations and management symposium workshops (NOMS Wksps* (pp. 88–94).

Solnon, C., Cung, V.-D., Nguyen, A., & Artigues, C. (2008). The car sequencing problem: Overview of state-of-the-art methods and industrial case-study of the ROADEF'2005 challenge problem. *European Journal of Operational Research*, *191*(3), 912–927.

Wang, X., Wang, Y., & Cui, Y. (2014). A new multi-objective bi-level programming model for energy and locality aware multi-job scheduling in cloud computing. *Future Generation Computer Systems*, *36*, 91–101.

Xu, F., Liu, F., Jin, H., & Vasilakos, A. V. (2014). Managing performance overhead of virtual machines in cloud computing: A survey, state of the art, and future directions. *Proceedings of the IEEE*, *102*(1), 11–31.

Yang, H., & Tate, M. (2012). A descriptive literature review and classification of cloud computing research. *Communications of the Association for Information Systems*, *31*(2), 35–60.

Zhao, L., Sakr, S., Liu, A., & Bouguettaya, A. (2014). Cloud computing. In *Cloud data management* (pp. 9–20). Berlin: Springer.