International Journal of Automation and Computing
9(4), August 2012, 429-441
DOI: 10.1007/s11633-012-0664-y

Task-resource Scheduling Problem

Anna Gorbenko Vladimir Popov
Department of Intelligent Systems and Robotics, Ural State University, Ekaterinburg 620083, Russian Federation

Abstract: Cloud computing is a new and rapidly emerging computing paradigm where applications, data and IT services are provided over the Internet. The task-resource management is the key role in cloud computing systems. Task-resource scheduling problems are premier which relate to the efficiency of the whole cloud computing facilities. Task-resource scheduling problem is NP-complete. In this paper, we consider an approach to solve this problem optimally. This approach is based on constructing a logical model for the problem. Using this model, we can apply algorithms for the satisfiability problem (SAT) to solve the task-resource scheduling problem. Also, this model allows us to create a testbed for particle swarm optimization algorithms for scheduling workflows.

Keywords: Clouds, complexity theory, scheduling, satisfiability problem, genetic algorithms.

1 Introduction

Problems of cloud computing are among the most rapidly developing areas of modern computer science[1−4]. Cloud computing is a rapidly emerging paradigm for distributed computing that delivers infrastructure, platform, and software as services. Such services are made available as subscription-based services in a pay-as-you-go model to consumers[5, 6]. When the user application requires computing resources, cloud computing helps user applications dynamically provision as many computing resources at specified locations as required. Usually, scientific workflows need to process huge amount of data and computationally intensive activities. Scientific workflow management systems are used for managing these scientific experiments by hiding the orchestration and inherent integration details while executing workflows on distributed resources provided by cloud service providers[7]. In this paper, we consider a problem of minimizing the total execution cost of applications on these resources provided by cloud service providers, such as Amazon and GoGrid[8, 9]. In particular, we consider the task-resource scheduling problem. This problem is NP-complete[10]. However, past work has proposed many heuristics based approaches to scheduling workflow applications[11−35]. In particular, genetic algorithms[12, 18, 19, 22, 27, 31−33, 35], simulated annealing[24, 30], tabu search[14], stochastic integer programming[16], and particle swarm optimization algorithms[25, 26, 28, 34] have been used for scheduling workflows. A good survey of such workflow scheduling algorithms for grid computing is given in [33]. Note that in several works, some relatively simple models of schedulers are considered. For instance, an approach based on genetic algorithm for scheduling only decomposable data grid applications is considered in [19]. Two models based on genetic algorithm for predicting only the completion time of jobs in a service grid are proposed in [18]. However, in most studies, multi-objective problems are considered. For example, security-driven heuristics and a fast genetic algorithm are proposed in [27]. The design, implementation and test results for a scheduler based on genetic algorithm are presented in [12]. The scheduler allows to minimize make-span, idle time of the available computational resources, turn-around time and the specified deadlines provided by users. Some schedulers based on genetic algorithm for heterogeneous computing environments are considered in [22, 23]. In this paper, we consider one of the most recent and general models which was proposed in [25]. Note that heuristic approaches[11−35] to scheduling workflow applications give us only suboptimal algorithms. Consequently, exhaustive searches were used to verify the quality of such approaches[31].

In this paper, we describe an approach to solve this problem optimally. In Section 2, we consider the problem of finding a task-resource mapping. In particular, we consider instances such that the highest cost among all the computing resources is minimized. Also, we consider instances such that the cost of all the computing resources is minimized. For these problems, we construct reductions to satisfiability problem and 3-satisfiability problem in Section 3. Data for computational experiments is considered in Section 4. In Section 5, we propose a SAT solver based on genetic algorithm. This algorithm allows us to solve considered scheduling problems optimally. In Section 6, we use this algorithm as the testbed for particle swarm optimization algorithm.

2 Problem definition

We can consider an application workflow as a directed acyclic graph represented by $G = (V, E)$, where

$$V = \{T_1, T_2, \cdots, T_n\}$$

is the set of tasks, and

$$E \subseteq \{(T_i, T_j) \mid T_i, T_j \in V\}$$

represents the data dependencies between these tasks. In particular, we suppose that $(T_i, T_j) \in E$ if and only if the data is produced by T_i and consumed by T_j. Suppose that

$$\mathcal{P} = \{P_1, P_2, \cdots, P_q\}$$

is a set of compute sites. We can suppose that the average computation time of a task T_i on a computing resource P_j

Manuscript received May 24, 2011; revised January 11, 2012
The work was partially supported by Analytical Departmental Program "Developing the Scientific Potential of Higher School" (Nos. 2.1.1/14055 and 2.1.1/13995).

for a certain size of input is known. In this case, the cost of computation of a task on a compute host is inversely proportional to the time it takes for computation on that resource. Let $w[i,j]$ be the cost of computation of a task T_i on a compute host P_j. Also, we can suppose that the cost of unit data access $d[i,j]$ from a resource P_i to a resource P_j is known. The cost of access is fixed by the service provider. Note that the transfer cost can be calculated according to the bandwidth between the sites. However, following [25], we have used the cost for transferring unit data between sites, per second. Also, following [25], we assume that these costs are non-negative, symmetric, and satisfy the triangle inequality:

$$d[i,j] \geqslant 0$$

$$d[i,j] = d[j,i]$$

$$d[i,j] + d[j,l] \geqslant d[i,l]$$

for all i, j, l. Suppose that two tasks T_{i_1} and T_{i_2} have file dependency between them. Let $e[i_1,i_2]$ be the output file size from T_{i_1} to T_{i_2}. Let M be a task-resource mapping instance.

For a given assignment M, the total cost $\mathcal{CT}_j(M)$ for a computing resource P_j is the sum of execution cost $\mathcal{CE}_j(M)$ and access cost $\mathcal{CA}_j(M)$.

Now, the problem of task-resource scheduling can be stated as:

TRS_HCR: Find a task-resource mapping instance M, such that when estimating the total cost incurred using each computing resource P_j, the highest cost among all the computing resources is minimized.

Subsequent minimization of the overall cost is

$$\mathcal{MH} = \min_M \max_{P_j} \mathcal{CT}_j(M)$$

where

$$\mathcal{CT}_j(M) = \mathcal{CE}_j(M) + \mathcal{CA}_j(M)$$

$$\mathcal{CA}_j(M) = \sum_{T_s \in V, T_t \in V, M(T_s)=P_j, M(T_t)=P_l \neq P_j} d[j,l]e[s,t]$$

$$\mathcal{CE}_j(M) = \sum_{M(T_s)=P_j} w[s,j].$$

This minimization ensures that the total cost is minimal even after initial distribution.

We also consider the following problem:

TRS_OCR: Find a task-resource mapping instance M, such that when estimating the total cost incurred using each computing resource P_j, the cost of all the computing resources is minimized.

Subsequent minimization of the overall cost

$$\mathcal{MO} = \min_M \sum_{P_j} \mathcal{CT}_j(M).$$

3 Logical models of TRS_HCR and TRS_OCR

The problem SAT is to determine whether the variables of a given Boolean function in conjunctive normal form (CNF) have an assignment that makes the function "true". Different variants of SAT were considered. Note that the problem SAT remains NP-complete even if all expressions are written in conjunctive normal form with 3 variables per clause (3-CNF). The problem 3SAT is to determine whether the variables of a given 3-CNF have an assignment that makes the function "true".

The satisfiability problem is fundamental in solving many problems in automated reasoning, computer-aided design, computer-aided manufacturing, machine vision, database, robotics, integrated circuit design, computer architecture design, and computer network design. In recent years, many optimization methods, parallel algorithms, and practical techniques have been developed for solving the satisfiability problem[36].

It is natural to use a reduction to different variants of the satisfiability problem to solve computationally hard problems. Encoding problems as Boolean satisfiability and solving them with very efficient satisfiability algorithms have recently caused considerable interest. There are several ways of SAT-encoding constraint satisfaction[37−40], clique[41], planning[42−44], some versions of scheduling[45−49], coloring[41, 50], the Hamiltonian cycle[41, 50], and some robotic problems[51−54].

A toolbox for Matlab TORSCHE Scheduling[55, 56] contains a number of scheduling algorithms. TORSCHE Scheduling deals with scheduling on monoprocessor, dedicated processors, parallel processors and with cyclic scheduling. In TORSCHE Scheduling, scheduling algorithms are categorized by notation α, β, and γ[57, 58]. In particular, in TORSCHE Scheduling, the SAT based approach to the scheduling problems is considered. In the toolbox a zChaff, SAT solver[59] is used to decide whether the set of clauses is satisfiable. If it is satisfiable, the schedule within s time units is feasible. After this, an optimal schedule is found in iterative manner. The list scheduling algorithm is used to find the initial value of s. Then value of s is iteratively decremented by one and feasibility of the solution is tested. When the solution is not feasible, the iterative algorithm finishes. Note that zChaff can be used to solve the problems with more than one million variables and 10 million clauses.

In this paper, we consider reductions from TRS_HCR and TRS_OCR to SAT and 3SAT.

The decision version of TRS_HCR can be formulated as following.

TRS_HCR_D:

Instance: Given a positive integer R, nonnegative integers $d[j,l]$, $e[s,t]$, $w[s,j]$, a set $\mathcal{P} = \{P_1, P_2, \cdots, P_q\}$, and a directed acyclic graph represented by $G = (V, E)$, where

$$V = \{T_1, T_2, \cdots, T_n\}$$

$$E \subseteq \{(T_i, T_j) \mid T_i, T_j \in V\}$$

$$1 \leqslant j \leqslant q, 1 \leqslant l \leqslant q, 1 \leqslant s \leqslant n, 1 \leqslant t \leqslant n.$$

Question: Is there a mapping $M : T \to \mathcal{P}$ such that

$$\max_{P_j} \mathcal{CT}_j(M) \leqslant R\ ?$$

Respectively, the decision version of TRS_OCR can be formulated as follows.

TRS_OCR_D:

Instance: Given an instance of TRS_HCR_D.
Question: Is there a mapping $M : T \to \mathcal{P}$ such that

$$\sum_{P_j} \mathcal{CT}_j(M) \leqslant R\ ?$$

Let

$$r_1 = \max_{1 \leqslant s \leqslant n, 1 \leqslant j \leqslant q} \lceil \log w[s,j] \rceil + 1$$

$$r_2 = \max_{1 \leqslant j \leqslant q, 1 \leqslant l \leqslant q} \lceil \log d[j,l] \rceil + 1$$

$$r_3 = \max_{1 \leqslant s \leqslant n, 1 \leqslant t \leqslant n} \lceil \log e[s,t] \rceil + 1$$

$$r = (n^2 + q^2) \max\{r_1, r_2 + r_3\}.$$

Suppose that

$$w[s,j] = a[s,j,r]a[s,j,r-1] \cdots a[s,j,1],$$

$$d[j,l]e[s,t] = b[j,l,s,t,r]b[j,l,s,t,r-1] \cdots b[j,l,s,t,1],$$

$$R = R[r]R[r-1] \cdots R[1]$$

where

$$1 \leqslant j \leqslant q, 1 \leqslant l \leqslant q,$$

$$1 \leqslant s \leqslant n, 1 \leqslant t \leqslant n,$$

$$a[s,j,c] \in \{0,1\}, b[j,l,s,t,c] \in \{0,1\},$$

$$R[c] \in \{0,1\}, 1 \leqslant c \leqslant r.$$

Let

$$\varphi[s,1] = \vee_{1 \leqslant j \leqslant q} x[s,j],$$

$$\varphi[s,2] = \wedge_{1 \leqslant j[1] < j[2] \leqslant q} (\neg x[s,j[1]] \vee \neg x[s,j[2]]),$$

$$\varphi = \wedge_{1 \leqslant s \leqslant n} (\varphi[s,1] \wedge \varphi[s,2]),$$

$$\delta[1] = \wedge_{1 \leqslant j \leqslant q, 1 \leqslant c \leqslant r} \neg y_1[0,j,c],$$

$$\delta[2] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant j \leqslant q} \neg u_1[s,j,0],$$

$$\psi[1,j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r} (x[s,j] \to ((y_1[s-1,j,c] \wedge a[s,j,c] \wedge u_1[s,j,c-1]) \to (y_1[s,j,c] \wedge u_1[s,j,c]))),$$

$$\psi[2,j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r} (x[s,j] \to (\neg y_1[s-1,j,c] \wedge a[s,j,c] \wedge u_1[s,j,c-1]) \to (\neg y_1[s,j,c] \wedge u_1[s,j,c]))),$$

$$\psi[3,j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r} (x[s,j] \to ((y_1[s-1,j,c] \wedge \neg a[s,j,c] \wedge u_1[s,j,c-1]) \to (\neg y_1[s,j,c] \wedge u_1[s,j,c]))),$$

$$\psi[4,j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r} (x[s,j] \to ((y_1[s-1,j,c] \wedge a[s,j,c] \wedge \neg u_1[s,j,c-1]) \to (\neg y_1[s,j,c] \wedge u_1[s,j,c]))),$$

$$\psi[5,j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r} (x[s,j] \to ((y_1[s-1,j,c] \wedge \neg a[s,j,c] \wedge \neg u_1[s,j,c-1]) \to (y_1[s,j,c] \wedge \neg u_1[s,j,c]))),$$

$$\psi[6,j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r} (x[s,j] \to ((\neg y_1[s-1,j,c] \wedge a[s,j,c] \wedge \neg u_1[s,j,c-1]) \to (y_1[s,j,c] \wedge \neg u_1[s,j,c]))),$$

$$\psi[7,j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r} (x[s,j] \to ((\neg y_1[s-1,j,c] \wedge \neg a[s,j,c] \wedge u_1[s,j,c-1]) \to (y_1[s,j,c] \wedge \neg u_1[s,j,c]))),$$

$$\psi[8,j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r} (x[s,j] \to ((\neg y_1[s-1,j,c] \wedge \neg a[s,j,c] \wedge \neg u_1[s,j,c-1]) \to (\neg y_1[s,j,c] \wedge \neg u_1[s,j,c]))),$$

$$\psi[9,j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r} (\neg x[s,j] \to ((y_1[s-1,j,c] \wedge u_1[s,j,c-1]) \to (\neg y_1[s,j,c] \wedge u_1[s,j,c]))),$$

$$\psi[10,j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r} (\neg x[s,j] \to ((\neg y_1[s-1,j,c] \wedge u_1[s,j,c-1]) \to (y_1[s,j,c] \wedge \neg u_1[s,j,c]))),$$

$$\psi[11,j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r} (\neg x[s,j] \to ((y_1[s-1,j,c] \wedge \neg u_1[s,j,c-1]) \to (y_1[s,j,c] \wedge \neg u_1[s,j,c]))),$$

$$\psi[12,j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r} (\neg x[s,j] \to ((\neg y_1[s-1,j,c] \wedge \neg u_1[s,j,c-1]) \to (\neg y_1[s,j,c] \wedge \neg u_1[s,j,c]))),$$

$$\psi[j] = \wedge_{i=1}^{12} \psi[i,j],$$

$$\psi = \wedge_{1 \leqslant j \leqslant q} \psi[j],$$

$$\delta[3] = \wedge_{1 \leqslant c \leqslant r} \neg y_2[n,0,c],$$

$$\delta[4] = \wedge_{1 \leqslant j \leqslant q} \neg u_2[n,j,0],$$

$$\varepsilon[1] = \wedge_{1 \leqslant j \leqslant q, 1 \leqslant c \leqslant r} ((y_2[n,j-1,c] \wedge y_1[n,j,c] \wedge u_2[n,j,c-1]) \to (y_2[n,j,c] \wedge u_2[n,j,c])),$$

$$\varepsilon[2] = \wedge_{1 \leqslant j \leqslant q, 1 \leqslant c \leqslant r} ((\neg y_2[n,j-1,c] \wedge y_1[n,j,c] \wedge u_2[n,j,c-1]) \to (\neg y_2[n,j,c] \wedge u_2[n,j,c])),$$

$$\varepsilon[3] = \wedge_{1 \leqslant j \leqslant q, 1 \leqslant c \leqslant r} ((y_2[n,j-1,c] \wedge \neg y_1[n,j,c] \wedge u_2[n,j,c-1]) \to (\neg y_2[n,j,c] \wedge u_2[n,j,c])),$$

$$\varepsilon[4] = \wedge_{1 \leqslant j \leqslant q, 1 \leqslant c \leqslant r} ((y_2[n,j-1,c] \wedge y_1[n,j,c] \wedge \neg u_2[n,j,c-1]) \to (\neg y_2[n,j,c] \wedge u_2[n,j,c])),$$

$$\varepsilon[5] = \wedge_{1 \leqslant j \leqslant q, 1 \leqslant c \leqslant r} ((\neg y_2[n,j-1,c] \wedge \neg y_1[n,j,c] \wedge u_2[n,j,c-1]) \to (y_2[n,j,c] \wedge \neg u_2[n,j,c])),$$

$$\varepsilon[6] = \wedge_{1\leqslant j\leqslant q,1\leqslant c\leqslant r}((\neg y_2[n,j-1,c]\wedge y_1[n,j,c]\wedge\neg u_2[n,j,c-1])\rightarrow (y_2[n,j,c]\wedge\neg u_2[n,j,c])),$$

$$\varepsilon[7] = \wedge_{1\leqslant j\leqslant q,1\leqslant c\leqslant r}((y_2[n,j-1,c]\wedge \neg y_1[n,j,c]\wedge\neg u_2[n,j,c-1])\rightarrow (y_2[n,j,c]\wedge\neg u_2[n,j,c])),$$

$$\varepsilon[8] = \wedge_{1\leqslant j\leqslant q,1\leqslant c\leqslant r}((\neg y_2[n,j-1,c]\wedge \neg y_1[n,j,c]\wedge\neg u_2[n,j,c-1])\rightarrow (\neg y_2[n,j,c]\wedge\neg u_2[n,j,c])),$$

$$\varepsilon = \wedge_{i=1}^{8}\varepsilon[i],$$

$$\delta[5] = \wedge_{1\leqslant j\leqslant q,1\leqslant l\leqslant q,1\leqslant s\leqslant n,1\leqslant c\leqslant r}\neg z_1[j,l,s,0,c],$$

$$\delta[6] = \wedge_{1\leqslant j\leqslant q,1\leqslant l\leqslant q,1\leqslant s\leqslant n,1\leqslant t\leqslant n}\neg v_1[j,l,s,t,0],$$

$$\rho[1,j,s,l] = \wedge_{1\leqslant t\leqslant n,l\neq j,1\leqslant c\leqslant r}((x[s,j]\wedge x[t,l])\rightarrow ((z_1[j,l,s,t-1,c]\wedge b[j,l,s,t,c]\wedge v_1[j,l,s,t,c-1])\rightarrow (z_1[j,l,s,t,c]\wedge v_1[j,l,s,t,c]))),$$

$$\rho[2,j,s,l] = \wedge_{1\leqslant t\leqslant n,l\neq j,1\leqslant c\leqslant r}((x[s,j]\wedge x[t,l])\rightarrow ((\neg z_1[j,l,s,t-1,c]\wedge b[j,l,s,t,c]\wedge v_1[j,l,s,t,c-1])\rightarrow (\neg z_1[j,l,s,t,c]\wedge v_1[j,l,s,t,c]))),$$

$$\rho[3,j,s,l] = \wedge_{1\leqslant t\leqslant n,l\neq j,1\leqslant c\leqslant r}((x[s,j]\wedge x[t,l])\rightarrow ((z_1[j,l,s,t-1,c]\wedge \neg b[j,l,s,t,c]\wedge v_1[j,l,s,t,c-1])\rightarrow (\neg z_1[j,l,s,t,c]\wedge v_1[j,l,s,t,c]))),$$

$$\rho[4,j,s,l] = \wedge_{1\leqslant t\leqslant n,l\neq j,1\leqslant c\leqslant r}((x[s,j]\wedge x[t,l])\rightarrow ((z_1[j,l,s,t-1,c]\wedge b[j,l,s,t,c]\wedge \neg v_1[j,l,s,t,c-1])\rightarrow (\neg z_1[j,l,s,t,c]\wedge v_1[j,l,s,t,c]))),$$

$$\rho[5,j,s,l] = \wedge_{1\leqslant t\leqslant n,l\neq j,1\leqslant c\leqslant r}((x[s,j]\wedge x[t,l])\rightarrow ((\neg z_1[j,l,s,t-1,c]\wedge \neg b[j,l,s,t,c]\wedge v_1[j,l,s,t,c-1])\rightarrow (z_1[j,l,s,t,c]\wedge \neg v_1[j,l,s,t,c]))),$$

$$\rho[6,j,s,l] = \wedge_{1\leqslant t\leqslant n,l\neq j,1\leqslant c\leqslant r}((x[s,j]\wedge x[t,l])\rightarrow ((\neg z_1[j,l,s,t-1,c]\wedge b[j,l,s,t,c]\wedge \neg v_1[j,l,s,t,c-1])\rightarrow (z_1[j,l,s,t,c]\wedge \neg v_1[j,l,s,t,c]))),$$

$$\rho[7,j,s,l] = \wedge_{1\leqslant t\leqslant n,l\neq j,1\leqslant c\leqslant r}((x[s,j]\wedge x[t,l])\rightarrow ((z_1[j,l,s,t-1,c]\wedge \neg b[j,l,s,t,c]\wedge \neg v_1[j,l,s,t,c-1])\rightarrow (z_1[j,l,s,t,c]\wedge \neg v_1[j,l,s,t,c]))),$$

$$\rho[8,j,s,l] = \wedge_{1\leqslant t\leqslant n,l\neq j,1\leqslant c\leqslant r}((x[s,j]\wedge x[t,l])\rightarrow ((\neg z_1[j,l,s,t-1,c]\wedge \neg b[j,l,s,t,c]\wedge \neg v_1[j,l,s,t,c-1])\rightarrow (\neg z_1[j,l,s,t,c]\wedge \neg v_1[j,l,s,t,c]))),$$

$$\rho[9,j,s,l] = \wedge_{1\leqslant t\leqslant n,1\leqslant c\leqslant r}((\neg x[s,j]\vee \neg x[t,l])\rightarrow ((z_1[j,l,s,t-1,c]\wedge v_1[j,l,s,t,c-1])\rightarrow (\neg z_1[j,l,s,t,c]\wedge v_1[j,l,s,t,c]))),$$

$$\rho[10,j,s,l] = \wedge_{1\leqslant t\leqslant n,1\leqslant c\leqslant r}((\neg x[s,j]\vee \neg x[t,l])\rightarrow ((\neg z_1[j,l,s,t-1,c]\wedge v_1[j,l,s,t,c-1])\rightarrow (z_1[j,l,s,t,c]\wedge \neg v_1[j,l,s,t,c]))),$$

$$\rho[11,j,s,l] = \wedge_{1\leqslant t\leqslant n,1\leqslant c\leqslant r}((\neg x[s,j]\vee \neg x[t,l])\rightarrow ((z_1[j,l,s,t-1,c]\wedge \neg v_1[j,l,s,t,c-1])\rightarrow (z_1[j,l,s,t,c]\wedge \neg v_1[j,l,s,t,c]))),$$

$$\rho[12,j,s,l] = \wedge_{1\leqslant t\leqslant n,1\leqslant c\leqslant r}((\neg x[s,j]\vee \neg x[t,l])\rightarrow ((\neg z_1[j,l,s,t-1,c]\wedge \neg v_1[j,l,s,t,c-1])\rightarrow (\neg z_1[j,l,s,t,c]\wedge \neg v_1[j,l,s,t,c]))),$$

$$\rho[13,j,s,l] = \wedge_{1\leqslant t\leqslant n,l=j,1\leqslant c\leqslant r}((x[s,j]\wedge x[t,l])\rightarrow ((z_1[j,l,s,t-1,c]\wedge v_1[j,l,s,t,c-1])\rightarrow (\neg z_1[j,l,s,t,c]\wedge v_1[j,l,s,t,c]))),$$

$$\rho[14,j,s,l] = \wedge_{1\leqslant t\leqslant n,l=j,1\leqslant c\leqslant r}((x[s,j]\wedge x[t,l])\rightarrow ((\neg z_1[j,l,s,t-1,c]\wedge v_1[j,l,s,t,c-1])\rightarrow (z_1[j,l,s,t,c]\wedge \neg v_1[j,l,s,t,c]))),$$

$$\rho[15,j,s,l] = \wedge_{1\leqslant t\leqslant n,l=j,1\leqslant c\leqslant r}((x[s,j]\wedge x[t,l])\rightarrow ((z_1[j,l,s,t-1,c]\wedge \neg v_1[j,l,s,t,c-1])\rightarrow (z_1[j,l,s,t,c]\wedge \neg v_1[j,l,s,t,c]))),$$

$$\rho[16,j,s,l] = \wedge_{1\leqslant t\leqslant n,l=j,1\leqslant c\leqslant r}((x[s,j]\wedge x[t,l])\rightarrow ((\neg z_1[j,l,s,t-1,c]\wedge \neg v_1[j,l,s,t,c-1])\rightarrow (\neg z_1[j,l,s,t,c]\wedge \neg v_1[j,l,s,t,c]))),$$

$$\delta[7] = \wedge_{1\leqslant j\leqslant q,1\leqslant s\leqslant n,1\leqslant c\leqslant r}\neg z_2[j,0,s,n,c],$$

$$\delta[8] = \wedge_{1\leqslant j\leqslant q,1\leqslant l\leqslant q,1\leqslant s\leqslant n}\neg v_2[j,l,s,n,0],$$

$$\rho[17,j,s] = \wedge_{1\leqslant l\leqslant q,l\neq j,1\leqslant c\leqslant r}(x[s,j]\rightarrow ((z_2[j,l-1,s,n,c]\wedge z_1[j,l,s,n,c]\wedge v_2[j,l,s,n,c-1])\rightarrow (z_2[j,l,s,n,c]\wedge v_2[j,l,s,n,c]))),$$

$$\rho[18,j,s] = \wedge_{1\leqslant l\leqslant q,l\neq j,1\leqslant c\leqslant r}(x[s,j]\rightarrow ((\neg z_2[j,l-1,s,n,c]\wedge z_1[j,l,s,n,c]\wedge v_2[j,l,s,n,c-1])\rightarrow (\neg z_2[j,l,s,n,c]\wedge v_2[j,l,s,n,c]))),$$

$$\rho[19,j,s] = \wedge_{1\leqslant l\leqslant q,l\neq j,1\leqslant c\leqslant r}(x[s,j]\rightarrow ((z_2[j,l-1,s,n,c]\wedge \neg z_1[j,l,s,n,c]\wedge v_2[j,l,s,n,c-1])\rightarrow (\neg z_2[j,l,s,n,c]\wedge v_2[j,l,s,n,c]))),$$

$$\rho[20,j,s] = \wedge_{1\leqslant l\leqslant q,l\neq j,1\leqslant c\leqslant r}(x[s,j]\rightarrow ((z_2[j,l-1,s,n,c]\wedge z_1[j,l,s,n,c]\wedge \neg v_2[j,l,s,n,c-1])\rightarrow (\neg z_2[j,l,s,n,c]\wedge v_2[j,l,s,n,c]))),$$

$$\rho[21,j,s] = \wedge_{1\leqslant l\leqslant q,l\neq j,1\leqslant c\leqslant r}(x[s,j]\rightarrow ((\neg z_2[j,l-1,s,n,c]\wedge \neg z_1[j,l,s,n,c]\wedge v_2[j,l,s,n,c-1])\rightarrow (z_2[j,l,s,n,c]\wedge \neg v_2[j,l,s,n,c]))),$$

$\rho[22, j, s] = \wedge_{1 \leqslant l \leqslant q, l \neq j, 1 \leqslant c \leqslant r}(x[s, j] \rightarrow$
$((\neg z_2[j, l-1, s, n, c] \wedge z_1[j, l, s, n, c] \wedge \neg v_2[j, l, s, n, c-1]) \rightarrow$
$(z_2[j, l, s, n, c] \wedge \neg v_2[j, l, s, n, c]))),$

$\rho[23, j, s] = \wedge_{1 \leqslant l \leqslant q, l \neq j, 1 \leqslant c \leqslant r}(x[s, j] \rightarrow$
$((z_2[j, l-1, s, n, c] \wedge \neg z_1[j, l, s, n, c] \wedge \neg v_2[j, l, s, n, c-1]) \rightarrow$
$(z_2[j, l, s, n, c] \wedge \neg v_2[j, l, s, n, c]))),$

$\rho[24, j, s] = \wedge_{1 \leqslant l \leqslant q, l \neq j, 1 \leqslant c \leqslant r}(x[s, j] \rightarrow$
$((\neg z_2[j, l-1, s, n, c] \wedge \neg z_1[j, l, s, n, c] \wedge \neg v_2[j, l, s, n, c-1]) \rightarrow$
$(\neg z_2[j, l, s, n, c] \wedge \neg v_2[j, l, s, n, c]))),$

$\rho[25, j, s] = \wedge_{1 \leqslant l \leqslant q, l = j, 1 \leqslant c \leqslant r}(x[s, j] \rightarrow$
$((z_2[j, l-1, s, n, c] \wedge v_2[j, l, s, n, c-1]) \rightarrow$
$(\neg z_2[j, l, s, n, c] \wedge v_2[j, l, s, n, c]))),$

$\rho[26, j, s] = \wedge_{1 \leqslant l \leqslant q, l = j, 1 \leqslant c \leqslant r}(x[s, j] \rightarrow$
$((\neg z_2[j, l-1, s, n, c] \wedge v_2[j, l, s, n, c-1]) \rightarrow$
$(z_2[j, l, s, n, c] \wedge \neg v_2[j, l, s, n, c]))),$

$\rho[27, j, s] = \wedge_{1 \leqslant l \leqslant q, l = j, 1 \leqslant c \leqslant r}(x[s, j] \rightarrow$
$((z_2[j, l-1, s, n, c] \wedge \neg v_2[j, l, s, n, c-1]) \rightarrow$
$(z_2[j, l, s, n, c] \wedge \neg v_2[j, l, s, n, c]))),$

$\rho[28, j, s] = \wedge_{1 \leqslant l \leqslant q, l = j, 1 \leqslant c \leqslant r}(x[s, j] \rightarrow$
$((\neg z_2[j, l-1, s, n, c] \wedge \neg v_2[j, l, s, n, c-1]) \rightarrow$
$(\neg z_2[j, l, s, n, c] \wedge \neg v_2[j, l, s, n, c]))),$

$\rho[29, j, s] = \wedge_{1 \leqslant l \leqslant q, 1 \leqslant c \leqslant r}(\neg x[s, j] \rightarrow$
$((z_2[j, l-1, s, n, c] \wedge v_2[j, l, s, n, c-1]) \rightarrow$
$(\neg z_2[j, l, s, n, c] \wedge v_2[j, l, s, n, c]))),$

$\rho[30, j, s] = \wedge_{1 \leqslant l \leqslant q, 1 \leqslant c \leqslant r}(\neg x[s, j] \rightarrow$
$((\neg z_2[j, l-1, s, n, c] \wedge v_2[j, l, s, n, c-1]) \rightarrow$
$(z_2[j, l, s, n, c] \wedge \neg v_2[j, l, s, n, c]))),$

$\rho[31, j, s] = \wedge_{1 \leqslant l \leqslant q, 1 \leqslant c \leqslant r}(\neg x[s, j] \rightarrow$
$((z_2[j, l-1, s, n, c] \wedge \neg v_2[j, l, s, n, c-1]) \rightarrow$
$(z_2[j, l, s, n, c] \wedge \neg v_2[j, l, s, n, c]))),$

$\rho[32, j, s] = \wedge_{1 \leqslant l \leqslant q, 1 \leqslant c \leqslant r}(\neg x[s, j] \rightarrow$
$((\neg z_2[j, l-1, s, n, c] \wedge \neg v_2[j, l, s, n, c-1]) \rightarrow$
$(\neg z_2[j, l, s, n, c] \wedge \neg v_2[j, l, s, n, c]))),$

$$\delta[9] = \wedge_{1 \leqslant j \leqslant q, 1 \leqslant c \leqslant r} \neg z_3[j, q, 0, n, c],$$

$$\delta[10] = \wedge_{1 \leqslant j \leqslant q, 1 \leqslant s \leqslant n} \neg v_3[j, q, s, n, 0],$$

$\rho[33, j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r}(x[s, j] \rightarrow$
$((z_3[j, q, s-1, n, c] \wedge z_2[j, q, s, n, c] \wedge v_3[j, q, s, n, c-1]) \rightarrow$
$(z_3[j, q, s, n, c] \wedge v_3[j, q, s, n, c]))),$

$\rho[34, j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r}(x[s, j] \rightarrow$
$((\neg z_3[j, q, s-1, n, c] \wedge z_2[j, q, s, n, c] \wedge v_3[j, q, s, n, c-1]) \rightarrow$
$(\neg z_3[j, q, s, n, c] \wedge v_3[j, q, s, n, c]))),$

$\rho[35, j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r}(x[s, j] \rightarrow$
$((z_3[j, q, s-1, n, c] \wedge \neg z_2[j, q, s, n, c] \wedge v_3[j, q, s, n, c-1]) \rightarrow$
$(\neg z_3[j, q, s, n, c] \wedge v_3[j, q, s, n, c]))),$

$\rho[36, j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r}(x[s, j] \rightarrow$
$((z_3[j, q, s-1, n, c] \wedge z_2[j, q, s, n, c] \wedge \neg v_3[j, q, s, n, c-1]) \rightarrow$
$(\neg z_3[j, q, s, n, c] \wedge v_3[j, q, s, n, c]))),$

$\rho[37, j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r}(x[s, j] \rightarrow$
$((\neg z_3[j, q, s-1, n, c] \wedge \neg z_2[j, q, s, n, c] \wedge v_3[j, q, s, n, c-1]) \rightarrow$
$(z_3[j, q, s, n, c] \wedge \neg v_3[j, q, s, n, c]))),$

$\rho[38, j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r}(x[s, j] \rightarrow$
$((\neg z_3[j, q, s-1, n, c] \wedge z_2[j, q, s, n, c] \wedge \neg v_3[j, q, s, n, c-1]) \rightarrow$
$(z_3[j, q, s, n, c] \wedge \neg v_3[j, q, s, n, c]))),$

$\rho[39, j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r}(x[s, j] \rightarrow$
$((z_3[j, q, s-1, n, c] \wedge \neg z_2[j, q, s, n, c] \wedge \neg v_3[j, q, s, n, c-1]) \rightarrow$
$(z_3[j, q, s, n, c] \wedge \neg v_3[j, q, s, n, c]))),$

$\rho[40, j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r}(x[s, j] \rightarrow$
$((\neg z_3[j, q, s-1, n, c] \wedge \neg z_2[j, q, s, n, c] \wedge \neg v_3[j, q, s, n, c-1]) \rightarrow$
$(\neg z_3[j, q, s, n, c] \wedge \neg v_3[j, q, s, n, c]))),$

$\rho[41, j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r}(\neg x[s, j] \rightarrow$
$((z_3[j, q, s-1, n, c] \wedge v_3[j, q, s, n, c-1]) \rightarrow$
$(\neg z_3[j, q, s, n, c] \wedge v_3[j, q, s, n, c]))),$

$\rho[42, j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r}(\neg x[s, j] \rightarrow$
$((\neg z_3[j, q, s-1, n, c] \wedge v_3[j, q, s, n, c-1]) \rightarrow$
$(z_3[j, q, s, n, c] \wedge \neg v_3[j, q, s, n, c]))),$

$\rho[43, j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r}(\neg x[s, j] \rightarrow$
$((z_3[j, q, s-1, n, c] \wedge \neg v_3[j, q, s, n, c-1]) \rightarrow$
$(z_3[j, q, s, n, c] \wedge \neg v_3[j, q, s, n, c]))),$

$\rho[44, j] = \wedge_{1 \leqslant s \leqslant n, 1 \leqslant c \leqslant r}(\neg x[s, j] \rightarrow$
$((\neg z_3[j, q, s-1, n, c] \wedge \neg v_3[j, q, s, n, c-1]) \rightarrow$
$(\neg z_3[j, q, s, n, c] \wedge \neg v_3[j, q, s, n, c]))),$

$$\delta[11] = \wedge_{1 \leqslant c \leqslant r} \neg z_4[0, q, n, n, c],$$

$$\delta[12] = \wedge_{1 \leqslant j \leqslant q} \neg v_4[j, q, n, n, 0],$$

$\rho[45] = \wedge_{1 \leqslant j \leqslant q, 1 \leqslant c \leqslant r}((z_4[j-1, q, n, n, c] \wedge$
$z_3[j, q, n, n, c] \wedge v_4[j, q, n, n, c-1]) \rightarrow$
$(z_4[j, q, n, n, c] \wedge v_4[j, q, n, n, c])),$

$$\rho[46] = \wedge_{1\leqslant j\leqslant q,1\leqslant c\leqslant r}((\neg z_4[j-1,q,n,n,c]\wedge z_3[j,q,n,n,c] \wedge v_4[j,q,n,n,c-1]) \to (\neg z_4[j,q,n,n,c] \wedge v_4[j,q,n,n,c])),$$

$$\rho[47] = \wedge_{1\leqslant j\leqslant q,1\leqslant c\leqslant r}((z_4[j-1,q,n,n,c]\wedge \neg z_3[j,q,n,n,c] \wedge v_4[j,q,n,n,c-1]) \to (\neg z_4[j,q,n,n,c] \wedge v_4[j,q,n,n,c])),$$

$$\rho[48] = \wedge_{1\leqslant j\leqslant q,1\leqslant c\leqslant r}((z_4[j-1,q,n,n,c]\wedge z_3[j,q,n,n,c] \wedge \neg v_4[j,q,n,n,c-1]) \to (\neg z_4[j,q,n,n,c] \wedge v_4[j,q,n,n,c])),$$

$$\rho[49] = \wedge_{1\leqslant j\leqslant q,1\leqslant c\leqslant r}((\neg z_4[j-1,q,n,n,c]\wedge \neg z_3[j,q,n,n,c] \wedge v_4[j,q,n,n,c-1]) \to (z_4[j,q,n,n,c] \wedge \neg v_4[j,q,n,n,c])),$$

$$\rho[50] = \wedge_{1\leqslant j\leqslant q,1\leqslant c\leqslant r}((\neg z_4[j-1,q,n,n,c]\wedge z_3[j,q,n,n,c] \wedge \neg v_4[j,q,n,n,c-1]) \to (z_4[j,q,n,n,c] \wedge \neg v_4[j,q,n,n,c])),$$

$$\rho[51] = \wedge_{1\leqslant j\leqslant q,1\leqslant c\leqslant r}((z_4[j-1,q,n,n,c]\wedge \neg z_3[j,q,n,n,c] \wedge \neg v_4[j,q,n,n,c-1]) \to (z_4[j,q,n,n,c] \wedge \neg v_4[j,q,n,n,c])),$$

$$\rho[52] = \wedge_{1\leqslant j\leqslant q,1\leqslant c\leqslant r}((\neg z_4[j-1,q,n,n,c]\wedge \neg z_3[j,q,n,n,c] \wedge \neg v_4[j,q,n,n,c-1]) \to (\neg z_4[j,q,n,n,c] \wedge \neg v_4[j,q,n,n,c])),$$

$$\rho = (\wedge_{1\leqslant i\leqslant 16,1\leqslant s\leqslant n,1\leqslant j\leqslant q,1\leqslant l\leqslant q}\rho[i,j,s,l])\wedge (\wedge_{17\leqslant i\leqslant 32,1\leqslant s\leqslant n,1\leqslant j\leqslant q}\rho[i,j,s])\wedge (\wedge_{33\leqslant i\leqslant 44,1\leqslant j\leqslant q}\rho[i,j]) \wedge (\wedge_{45\leqslant i\leqslant 52}\rho[i]),$$

$$\delta[13] = \neg w[0],$$

$$\tau[1] = \wedge_{1\leqslant c\leqslant r}((y_2[n,q,c] \wedge z_4[q,q,n,n,c] \wedge w[c-1]) \to (z[c] \wedge w[c])),$$

$$\tau[2] = \wedge_{1\leqslant c\leqslant r}((\neg y_2[n,q,c] \wedge z_4[q,q,n,n,c] \wedge w[c-1]) \to (\neg z[c] \wedge w[c])),$$

$$\tau[3] = \wedge_{1\leqslant c\leqslant r}((y_2[n,q,c] \wedge \neg z_4[q,q,n,n,c] \wedge w[c-1]) \to (\neg z[c] \wedge w[c])),$$

$$\tau[4] = \wedge_{1\leqslant c\leqslant r}((y_2[n,q,c] \wedge z_4[q,q,n,n,c] \wedge \neg w[c-1]) \to (\neg z[c] \wedge w[c])),$$

$$\tau[5] = \wedge_{1\leqslant c\leqslant r}((\neg y_2[n,q,c] \wedge \neg z_4[q,q,n,n,c] \wedge w[c-1]) \to (z[c] \wedge \neg w[c])),$$

$$\tau[6] = \wedge_{1\leqslant c\leqslant r}((\neg y_2[n,q,c] \wedge z_4[q,q,n,n,c] \wedge \neg w[c-1]) \to (z[c] \wedge \neg w[c])),$$

$$\tau[7] = \wedge_{1\leqslant c\leqslant r}((y_2[n,q,c] \wedge \neg z_4[q,q,n,n,c] \wedge \neg w[c-1]) \to (z[c] \wedge \neg w[c])),$$

$$\tau[8] = \wedge_{1\leqslant c\leqslant r}((\neg y_2[n,q,c] \wedge \neg z_4[q,q,n,n,c] \wedge \neg w[c-1]) \to (\neg z[c] \wedge \neg w[c])),$$

$$\tau = \wedge_{i=1}^{8}\tau[i],$$

$$\delta[14] = R_1[r+1] \wedge \neg R_2[r+1],$$

$$\eta[1] = \wedge_{1\leqslant c\leqslant r}(R_2[c+1] \to R_2[c]),$$

$$\eta[2] = \wedge_{1\leqslant c\leqslant r}((R[c] \wedge \neg z[c] \wedge R_1[c+1]) \to R_2[c]),$$

$$\eta[3] = \wedge_{1\leqslant c\leqslant r}((R[c] \wedge z[c] \wedge R_1[c+1]) \to R_1[c]),$$

$$\eta[4] = \wedge_{1\leqslant c\leqslant r}((\neg R[c] \wedge \neg z[c] \wedge R_1[c+1]) \to R_1[c]),$$

$$\eta[5] = \wedge_{1\leqslant c\leqslant r}((\neg R[c] \wedge z[c]) \to \neg R_1[c]),$$

$$\eta[6] = \wedge_{1\leqslant c\leqslant r}(\neg R_1[c+1] \to \neg R_1[c]),$$

$$\eta = \wedge_{1\leqslant i\leqslant 6}\eta[i],$$

$$\delta = \wedge_{1\leqslant i\leqslant 14}\delta[i],$$

$$\xi_1 = \varphi \wedge \psi \wedge \varepsilon \wedge \rho \wedge \tau \wedge \eta \wedge \delta \wedge (R_1[1] \vee R_2[1]).$$

Theorem 1. Given an instance of TRS_OCR_D, there is a mapping $M : T \to \mathcal{P}$ such that

$$\sum_{P_j} \mathcal{CT}_j(M) \leqslant R$$

if and only if ξ_1 is satisfiable.

Proof. Suppose that there is a mapping $M : T \to \mathcal{P}$ such that $\sum_{P_j} \mathcal{CT}_j(M) \leqslant R$.

Let

$$x[s,j] = 1 \Leftrightarrow M(T_s) = P_j. \quad (1)$$

By definition, for any s, there is j such that $M(T_s) = P_j$. Respectively, for any s, there is j such that $x[s,j] = 1$. Therefore, $\varphi[s,1] = 1$ for any s. By definition, for any s, there is only one value of j such that $M(T_s) = P_j$. Respectively, for any s, there is only one value of j such that $x[s,j] = 1$. Therefore, $\varphi[s,2] = 1$ for any s. Since $\varphi[s,1] = 1$ and $\varphi[s,2] = 1$ for any s, it is clear that $\varphi = 1$.

Let $y_1[0,j,c] = 0$ and $u_1[s,j,0] = 0$ where

$$1 \leqslant j \leqslant q, \quad 1 \leqslant c \leqslant r, \quad 1 \leqslant s \leqslant n.$$

It is easy to see that $\delta[1] = 1$ and $\delta[2] = 1$.

Let

$$Y_1[s,j] = \sum_{M(T_a)=P_j, 1\leqslant f\leqslant s} w[f,j] \quad (2)$$

$$Y_1[s,j] = y_1[s,j,r]y_1[s,j,r-1]\cdots y_1[s,j,1], \quad (3)$$
$$y_1[s,j,c] \in \{0,1\},$$
$$1 \leqslant c \leqslant r.$$

Let

$$u_1[f,j,c] = 1 \quad (4)$$

if and only if

$$y_1[f-1,j,c]y_1[f-1,j,c-1]\cdots y_1[f-1,j,1]+ a[f,j,c]a[f,j,c-1]\cdots a[f,j,1]= y_1'[f,j,c+1]y_1[f,j,c]\cdots y_1[f,j,1] \tag{5}$$

where

$$y_1[s,j,f]\in\{0,1\},\quad 1\leqslant f\leqslant c,\quad y_1'[f,j,c+1]=1.$$

It is easy to check that $\psi=1$.

Suppose that $y_2[n,0,c]=0$, $u_2[n,j,0]=0$. It is clear that $\delta[3]=1$, $\delta[4]=1$. Let

$$y_2[n,j,r]y_2[n,j,r-1]\cdots y_2[n,j,1]=\sum_{1\leqslant f\leqslant j}\mathcal{CE}_f(M). \tag{6}$$

It is easy to check that $\varepsilon=1$.

Similarly, let

$$z_1[j,l,s,0,c]=0, v_1[j,l,s,t,0]=0,$$
$$z_2[j,0,s,n,c]=0, v_2[j,l,s,n,0]=0,$$
$$z_3[j,q,0,n,c]=0, v_3[j,q,s,n,0]=0,$$
$$z_4[0,q,n,n,c]=0, v_4[j,q,n,n,0]=0, w[0]=0,$$

$$z_1[j,l,s,t,r]z_1[j,l,s,t,r-1]\cdots z_1[j,l,s,t,1]= \sum_{1\leqslant f\leqslant t, T_s\in V, T_f\in V, M(T_s)=P_j, M(T_f)=P_l\neq P_j} d[j,l]e[s,f] \tag{7}$$

where $s=\text{const}$, $l=\text{const}$, $j=\text{const}$,

$$z_2[j,l,s,n,r]z_2[j,l,s,n,r-1]\cdots z_2[j,l,s,n,1]= \sum_{\substack{1\leqslant t\leqslant n, 1\leqslant f\leqslant l, T_s\in V, T_t\in V,\\ M(T_s)=P_j, M(T_t)=P_f\neq P_j}} d[j,f]e[s,t] \tag{8}$$

where $s=\text{const}$, $j=\text{const}$,

$$z_3[j,q,s,n,r]z_3[j,q,s,n,r-1]\cdots z_3[j,q,s,n,1]= \sum_{\substack{1\leqslant t\leqslant n, 1\leqslant l\leqslant q, 1\leqslant f\leqslant s, T_f\in V, T_t\in V,\\ M(T_f)=P_j, M(T_t)=P_l\neq P_j}} d[j,l]e[f,t] \tag{9}$$

where $j=\text{const}$,

$$z_4[j,q,n,n,r]z_4[j,q,n,n,r-1]\cdots z_4[j,q,n,n,1]= \sum_{1\leqslant f\leqslant j}\mathcal{CA}_f(M) \tag{10}$$

$$z[r]z[r-1]\cdots z[1]=\sum_{P_j}\mathcal{CT}_j(M). \tag{11}$$

Suppose that

$$v_1[j,l,s,f,c]=1 \tag{12}$$

if and only if

$$z_1[j,l,s,f-1,c]z_1[j,l,s,f-1,c-1]\cdots z_1[j,l,s,f-1,1]+ b[j,l,s,f,c]b[j,l,s,f,c-1]\cdots b[j,l,s,f,1]= z_1'[j,l,s,f,c+1]z_1[j,l,s,f,c]\cdots z_1[j,l,s,f,1] \tag{13}$$

where $z_1'[j,l,s,f,c+1]=1$;

$$v_2[j,f,s,n,c]=1 \tag{14}$$

if and only if

$$z_2[j,f-1,s,n,c]z_2[j,f-1,s,n,c-1]\cdots z_2[j,f-1,s,n,1]+ z_1[j,f,s,n,c]z_1[j,f,s,n,c-1]\cdots z_1[j,f,s,n,1]= z_2'[j,f,s,n,c+1]z_2[j,f,s,n,c]\cdots z_2[j,f,s,n,1] \tag{15}$$

where $z_2'[j,f,s,n,c+1]=1$;

$$v_3[j,q,f,n,c]=1 \tag{16}$$

if and only if

$$z_3[j,q,f-1,n,c]z_3[j,q,f-1,n,c-1]\cdots z_3[j,q,f-1,n,1]+ z_2[j,q,f,n,c]z_2[j,q,f,n,c-1]\cdots z_2[j,q,f,n,1]= z_3'[j,q,f,n,c+1]z_3[j,q,f,n,c]\cdots z_3[j,q,f,n,1] \tag{17}$$

where $z_3'[j,q,f,n,c+1]=1$;

$$v_4[f,q,n,n,c]=1 \tag{18}$$

if and only if

$$z_4[f-1,q,n,n,c]z_4[f-1,q,n,n,c-1]\cdots z_4[f-1,q,n,n,1]+ z_3[f,q,n,n,c]z_3[f,q,n,n,c-1]\cdots z_3[f,q,n,n,1]= z_4'[f,q,n,n,c+1]z_4[f,q,n,n,c]\cdots z_4[f,q,n,n,1] \tag{19}$$

where $z_4'[f,q,n,n,c+1]=1$;

$$w[c]=1 \tag{20}$$

if and only if

$$y_2[n,q,c]y_2[n,q,c-1]\cdots y_2[n,q,1]+ z_4[q,q,n,n,c]z_4[q,q,n,n,c-1]\cdots z_4[q,q,n,n,1]= z'[c+1]z[c]\cdots z[1] \tag{21}$$

where $z'[c+1]=1$. It is easy to check that in this case $\wedge_{1\leqslant i\leqslant 13}\delta[i]=1$, $\rho=1$, $\tau=1$.

Let $R_1[r+1]=1$ and $R_2[r+1]=0$. Clearly, $\delta[14]=1$. Since

$$\sum_{P_j}\mathcal{CT}_j(M)\leqslant R$$

it is easy to check that $\eta=1$. Note that $R_1[1]=1$ if

$$\sum_{P_j}\mathcal{CT}_j(M)=R.$$

Respectively, $R_2[1] = 1$ if

$$\sum_{P_j} \mathcal{CT}_j(M) < R.$$

Therefore, $\xi_1 = 1$.

Now suppose that $\xi_1 = 1$. In this case, (1) as a definition of mapping M can be used. Using relations (2)–(21), one can easily verify that

$$\sum_{P_j} \mathcal{CT}_j(M) \leqslant R.$$

□

Let

$$\delta[15] = \wedge_{1 \leqslant j \leqslant q} \neg p[0, j],$$

$$\vartheta[1, j] = \wedge_{1 \leqslant c \leqslant r}((y_1[n, j, c] \wedge z_3[j, q, n, n, c] \wedge p[c-1, j]) \to (z[c, j] \wedge p[c, j])),$$

$$\vartheta[2, j] = \wedge_{1 \leqslant c \leqslant r}((\neg y_1[n, j, c] \wedge z_3[j, q, n, n, c] \wedge p[c-1, j]) \to (\neg z[c, j] \wedge p[c, j])),$$

$$\vartheta[3, j] = \wedge_{1 \leqslant c \leqslant r}((y_1[n, j, c] \wedge \neg z_3[j, q, n, n, c] \wedge p[c-1, j]) \to (\neg z[c, j] \wedge p[c, j])),$$

$$\vartheta[4, j] = \wedge_{1 \leqslant c \leqslant r}((y_1[n, j, c] \wedge z_3[j, q, n, n, c] \wedge \neg p[c-1, j]) \to (\neg z[c, j] \wedge p[c, j])),$$

$$\vartheta[5, j] = \wedge_{1 \leqslant c \leqslant r}((\neg y_1[n, j, c] \wedge \neg z_3[j, q, n, n, c] \wedge p[c-1, j]) \to (z[c, j] \wedge \neg p[c, j])),$$

$$\vartheta[6, j] = \wedge_{1 \leqslant c \leqslant r}((\neg y_1[n, j, c] \wedge z_3[j, q, n, n, c] \wedge \neg p[c-1, j]) \to (z[c, j] \wedge \neg p[c, j])),$$

$$\vartheta[7, j] = \wedge_{1 \leqslant c \leqslant r}((y_1[n, j, c] \wedge \neg z_3[j, q, n, n, c] \wedge \neg p[c-1, j]) \to (z[c, j] \wedge \neg p[c, j])),$$

$$\vartheta[8, j] = \wedge_{1 \leqslant c \leqslant r}((\neg y_1[n, j, c] \wedge \neg z_3[j, q, n, n, c] \wedge \neg p[c-1, j]) \to (\neg z[c, j] \wedge \neg p[c, j])),$$

$$\vartheta = \wedge_{1 \leqslant i \leqslant 8, 1 \leqslant j \leqslant q} \vartheta[i, j],$$

$$\delta[16] = \wedge_{1 \leqslant j \leqslant q} R_1[r+1, j],$$

$$\delta[17] = \wedge_{1 \leqslant j \leqslant q} \neg R_2[r+1, j],$$

$$\eta[1, j] = \wedge_{1 \leqslant c \leqslant r}(R_2[c+1, j] \to R_2[c, j]),$$

$$\eta[2, j] = \wedge_{1 \leqslant c \leqslant r}((R[c] \wedge \neg z[c, j] \wedge R_1[c+1, j]) \to R_2[c, j]),$$

$$\eta[3, j] = \wedge_{1 \leqslant c \leqslant r}((R[c] \wedge z[c, j] \wedge R_1[c+1, j]) \to R_1[c, j]),$$

$$\eta[4, j] = \wedge_{1 \leqslant c \leqslant r}((\neg R[c] \wedge \neg z[c, j] \wedge R_1[c+1, j]) \to R_1[c, j]),$$

$$\eta[5, j] = \wedge_{1 \leqslant c \leqslant r}((\neg R[c] \wedge z[c, j]) \to \neg R_1[c, j]),$$

$$\eta[6, j] = \wedge_{1 \leqslant c \leqslant r}(\neg R_1[c+1, j] \to \neg R_1[c, j]),$$

$$\zeta = \wedge_{1 \leqslant i \leqslant 6, 1 \leqslant j \leqslant q} \eta[i, j],$$

$$\rho' = (\wedge_{1 \leqslant i \leqslant 16, 1 \leqslant s \leqslant n, 1 \leqslant j \leqslant q, 1 \leqslant l \leqslant q} \rho[i, j, s, l]) \wedge (\wedge_{17 \leqslant i \leqslant 32, 1 \leqslant s \leqslant n, 1 \leqslant j \leqslant q} \rho[i, j, s]) \wedge (\wedge_{33 \leqslant i \leqslant 44, 1 \leqslant j \leqslant q} \rho[i, j]),$$

$$\delta' = \delta[1] \wedge \delta[2] \wedge (\wedge_{5 \leqslant i \leqslant 10} \delta[i]) \wedge (\wedge_{15 \leqslant i \leqslant 17} \delta[i]),$$

$$\xi_2 = \varphi \wedge \psi \wedge \rho' \wedge \delta' \wedge \vartheta \wedge \zeta \wedge (\wedge_{1 \leqslant j \leqslant q}(R_1[1, j] \vee R_2[1, j])).$$

Theorem 2. Given an instance of TRS_HCR_D, there is a mapping $M : T \to \mathcal{P}$ such that

$$\max_{P_j} \mathcal{CT}_j(M) \leqslant R$$

if and only if ξ_2 is satisfiable.

Proof. Suppose that

$$\mathcal{CT}_j(M) = z[r, j] z[r-1, j] \cdots z[1, j] \tag{22}$$

where

$$z[c, j] \in \{0, 1\}, \quad 1 \leqslant c \leqslant r, \quad 1 \leqslant j \leqslant q.$$

Assume that

$$p[c, j] = 1 \tag{23}$$

if and only if

$$y_1[n, j, c] y_1[n, j, c-1] \cdots y_1[n, j, 1] + z_3[j, q, n, n, c] z_3[j, q, n, n, c-1] \cdots z_3[j, q, n, n, 1] = z'[c+1, j] z[c, j] \cdots z[1, j] \tag{24}$$

where $z'[c+1, j] = 1$. Based on assumptions (22)–(24), the proof of Theorem 2 is easily obtained using the same ideas as in the proof of Theorem 1. □

Note that

$$\beta \to \gamma = \neg\beta \vee \gamma \tag{25}$$

$$(\beta_1 \wedge \beta_2) \to \gamma = \neg(\beta_1 \wedge \beta_2) \vee \gamma = \neg\beta_1 \vee \neg\beta_2 \vee \gamma \tag{26}$$

$$(\beta_1 \wedge \beta_2 \wedge \beta_3) \to \gamma = \neg(\beta_1 \wedge \beta_2 \wedge \beta_3) \vee \gamma = \neg\beta_1 \vee \neg\beta_2 \vee \neg\beta_3 \vee \gamma. \tag{27}$$

It is easy to see that

$$\begin{aligned}(\beta_1 \wedge \beta_2 \wedge \beta_3) \to (\gamma_1 \wedge \gamma_2) = \\ \neg(\beta_1 \wedge \beta_2 \wedge \beta_3) \vee (\gamma_1 \wedge \gamma_2) = \\ \neg\beta_1 \vee \neg\beta_2 \vee \neg\beta_3) \vee (\gamma_1 \wedge \gamma_2) = \\ (\neg\beta_1 \vee \neg\beta_2 \vee \neg\beta_3 \vee \gamma_1) \wedge \\ (\neg\beta_1 \vee \neg\beta_2 \vee \neg\beta_3 \vee \gamma_2).\end{aligned} \tag{28}$$

Therefore,

$$\begin{aligned}\alpha \to ((\beta_1 \wedge \beta_2 \wedge \beta_3) \to (\gamma_1 \wedge \gamma_2)) = \\ \neg\alpha \vee ((\beta_1 \wedge \beta_2 \wedge \beta_3) \to (\gamma_1 \wedge \gamma_2)) = \\ (\neg\alpha \vee \neg\beta_1 \vee \neg\beta_2 \vee \neg\beta_3 \vee \gamma_1) \wedge \\ (\neg\alpha \vee \neg\beta_1 \vee \neg\beta_2 \vee \neg\beta_3 \vee \gamma_2).\end{aligned} \tag{29}$$

Similarly,

$$\begin{aligned}\alpha \to ((\beta_1 \wedge \beta_2) \to (\gamma_1 \wedge \gamma_2)) = \\ (\neg\alpha \vee \neg\beta_1 \vee \neg\beta_2 \vee \gamma_1) \wedge \\ (\neg\alpha \vee \neg\beta_1 \vee \neg\beta_2 \vee \gamma_2)\end{aligned} \tag{30}$$

$$(\alpha_1 \wedge \alpha_2) \to ((\beta_1 \wedge \beta_2 \wedge \beta_3) \to (\gamma_1 \wedge \gamma_2)) = \neg\alpha_1 \vee \neg\alpha_2 \vee ((\beta_1 \wedge \beta_2 \wedge \beta_3) \to (\gamma_1 \wedge \gamma_2)) = (\neg\alpha_1 \vee \neg\alpha_2 \vee \neg\beta_1 \vee \neg\beta_2 \vee \neg\beta_3 \vee \gamma_1) \wedge (\neg\alpha_1 \vee \neg\alpha_2 \vee \neg\beta_1 \vee \neg\beta_2 \vee \neg\beta_3 \vee \gamma_2) \quad (31)$$

$$(\alpha_1 \wedge \alpha_2) \to ((\beta_1 \wedge \beta_2) \to (\gamma_1 \wedge \gamma_2)) = \neg\alpha_1 \vee \neg\alpha_2 \vee ((\beta_1 \wedge \beta_2) \to (\gamma_1 \wedge \gamma_2)) = (\neg\alpha_1 \vee \neg\alpha_2 \vee \neg\beta_1 \vee \neg\beta_2 \vee \gamma_1) \wedge (\neg\alpha_1 \vee \neg\alpha_2 \vee \neg\beta_1 \vee \neg\beta_2 \vee \gamma_2) \quad (32)$$

Using relations (25)–(32), we can obtain explicit transformations of ξ_1 and ξ_2 into ξ'_1 and ξ'_2, respectively, such that $\xi_i \Leftrightarrow \xi'_i$ and τ is in CNF. Clearly, ξ'_1 and ξ'_2 give us explicit reductions from TRS_OCR_D and TRS_HCR_D, respectively, to SAT.

Note that

$$\alpha \Leftrightarrow (\alpha \vee \beta_1 \vee \beta_2) \wedge (\alpha \vee \neg\beta_1 \vee \beta_2) \wedge (\alpha \vee \beta_1 \vee \neg\beta_2) \wedge (\alpha \vee \neg\beta_1 \vee \neg\beta_2) \quad (33)$$

$$\vee_{j=1}^{l}\alpha_j \Leftrightarrow (\alpha_1 \vee \alpha_2 \vee \beta_1) \wedge (\wedge_{i=1}^{l-4}(\neg\beta_i \vee \alpha_{i+2} \vee \beta_{i+1})) \wedge (\neg\beta_{l-3} \vee \alpha_{l-1} \vee \alpha_l) \quad (34)$$

$$\alpha_1 \vee \alpha_2 \Leftrightarrow (\alpha_1 \vee \alpha_2 \vee \beta) \wedge (\alpha_1 \vee \alpha_2 \vee \neg\beta) \quad (35)$$

$$\vee_{j=1}^{4}\alpha_j \Leftrightarrow (\alpha_1 \vee \alpha_2 \vee \beta_1) \wedge (\neg\beta_1 \vee \alpha_3 \vee \alpha_4) \quad (36)$$

where $l > 4$. Using relations (33)–(36), we can obtain explicit transformations of ξ'_1 and ξ'_2 into ξ''_1 and ξ''_2, respectively, such that $\xi'_i \Leftrightarrow \xi''_i$ and τ is in 3CNF. Clearly, ξ''_1 and ξ''_2 give us explicit reductions from TRS_OCR_D and TRS_HCR_D, respectively, to 3SAT.

4 Data for experiments

Following [25], we have used three matrices that store the values for average computation cost of each task on each resource (TP-matrix), average communication cost per unit data between computing resources (PP-matrix), input/output data size of each task (DS-matrix). The values for PP-matrix resembling the cost of unit data transfer between resources were given by Amazon CloudFront[60]. While varying the processing cost, we use pricing policy from Amazon Elastic Compute Cloud[61] for different classes of virtual machine instances.

Also, we use enterprise cloud of the Department of Intelligent Systems and Robotics of Ural State University that is intended to solve robotics problems. The enterprise cloud of the department is composed of two clusters (8 calculation nodes, Intel Pentium IV 2.40 GHz processor, HDD 2×80 GB, Linux; 8 calculation nodes, Intel Pentium IV 2.40 GHz processor, HDD 2×80 GB, Windows XP SP2), 50 desktop personal computers (from Pentium IV 2.00 GHz to Pentium IV 3.40 GHz, from HDD 200 GB to HDD 500 GB, Linux, Windows XP SP2, Windows Vista, Windows 7), 80 monoblocs (Intel Atom 1.6 GHz processor, HDD 80 GB, Windows Vista), network storage system HP StorageWorks 48 TB, and 4 laptops used as scheduler nodes.

The cloud resources are located in three different student laboratories and seven different research laboratories. Also, the enterprise cloud of the department can use two clusters of Mathematics and Mechanics Institute of Ural Branch of Russian Academy of Sciences (umt, Linux, 1664 calculation nodes, Xeon 3.00 GHz processor; um64, Linux, 128 calculation nodes, AMD Opteron 2.6 GHz processor)[62].

The cloud is used to solve different tasks. In particular, it is used for large computational experiments[51–54, 63–66]. Also, it is used as an external computational resource of mobile robots. We use mobile robots with the following computational resources. Electronic systems: the ZX-SERVO 16 microcontroller or SSC-32 microcontroller, onboard computers: VIA processor, AMD Geode LX600 processor, Asus Eee PC 1000HE, Sony VAIO VPCS13X9R/B.

Robots have wireless access to resources of the cloud. Mobile robots use a recognition system which consists of separate recognition modules, intelligent system of selection of recognition modules, and intelligent generator of recognition modules. The recognition system uses recognition modules based on neural networks and threshold circuits. The intelligent system of selection of recognition modules considers each specific task and each particular environment. After this, the system determines which particular recognition module to be used for solving the current task.

If current task is new or the robot is in a new environment, then intelligent system of selection of recognition modules formulates the task of generation of a new module. The intelligent generator considers each new task and each new environment. After this, the generator produces new recognition module. Since onboard computing resources are very limited, the robot uses only one recognition module at each moment of time. The remaining part of the system is installed on the Cloud.

5 SAT solvers for TRS_OCR_D and TRS_HCR_D

In Section 3, we have obtained explicit reductions from TRS_OCR_D and TRS_HCR_D to SAT and 3SAT. We used the algorithms fgrasp and posit from SATLIB[67]. Also, we have designed our own genetic algorithm for SAT which is based on the algorithms from SATLIB[67].

Consider a Boolean function

$$g(x_1, x_2, \cdots, x_n) = \wedge_{i=1}^{m}\mathcal{C}_i$$

where $m \geqslant 1$, and each of the $\mathcal{C}_i$ is the disjunction of one or more literals. Let $|\mathcal{C}_i|$ be the number of literals in $\mathcal{C}_i$, $occ(x_i, g)$ be the number of occurrences of x_i in g, $occ(\neg x_i, g)$ be the number of occurrences of x_i in g, respectively. For example, if

$$g = (x_1 \vee x_2) \wedge (\neg x_2 \vee x_3) \wedge (x_1 \vee x_4) \wedge (\neg x_1 \vee x_5)$$

then $occ(x_1, g) = 2$, $occ(\neg x_1, g) = 1$.

We consider a number of natural principles that define the importance of a variable x_i for satisfiability of Boolean function g. These principles suggest us correct values of variables.

1) If $occ(x_i, g) \geqslant 0$ and $occ(\neg x_i, g) = 0$, then $x_i = 1$.
2) If $occ(x_i, g) = 0$ and $occ(\neg x_i, g) \geqslant 0$, then $x_i = 0$.
3) If $occ(x_i, g) > occ(\neg x_i, g)$, then $x_i = 1$.
4) If $occ(x_i, g) < occ(\neg x_i, g)$, then $x_i = 1$.
5) If $x_i = \mathcal{C}_j$ for some j, then $x_i = 1$.
6) If

$$\min_{occ(x_i,\mathcal{C}_j)>0} |\mathcal{C}_j| \leqslant \min_{occ(\neg x_i,\mathcal{C}_j)>0} |\mathcal{C}_j|$$

then $x_i = 1$.
7) Given positive integers

$$p_1, p_2, \cdots, p_m, q_1, q_2, \cdots, q_m$$

and a set of rational numbers

$$\{\alpha_{i,u}, \beta_{i,v} \mid 1 \leqslant i \leqslant m, 1 \leqslant u \leqslant p_i, 1 \leqslant v \leqslant q_i\}$$

if

$$\sum_{1\leqslant j\leqslant m, 1\leqslant u\leqslant p_j, |\mathcal{C}_j|=u} \alpha_{j,u} occ(x_i, \mathcal{C}_j) \geqslant \sum_{1\leqslant j\leqslant m, 1\leqslant v\leqslant q_j, |\mathcal{C}_j|=v} \beta_{j,v} occ(\neg x_i, \mathcal{C}_j)$$

then $x_i = 1$.

Based on these principles, we can consider the following seven types of commands: $P_1, P_2, \cdots, P_7$. Also, we consider the following three commands for running algorithms: Try_fgrasp, Try_posit, and Try_ga, where Try_ga runs a simple genetic algorithm which is similar to GASAT[68].

Denote $\mathcal{R}$ as the set of commands of these ten types. It is possible to consider arbitrary element of $\mathcal{R}^*$ as a program for finding the values of variables of a Boolean function. We assume that such programs are executed on a cluster.

Execution of each of commands of type P_i reduces the number of variables of a Boolean function by one. Execution of each of commands Try_fgrasp, Try_posit, and Try_ga consists in the run of corresponding algorithm for current Boolean function on a separate set of calculation nodes and the transition to the next command.

For algorithms fgrasp and posit, we only run on one calculation node. Genetic algorithms can be used in parallel execution. We use auxiliary genetic algorithm which determines the number of calculation nodes.

Initially, we selected a random subset of $\mathcal{R}^*$. We use a genetic algorithm to select a program from the current subset of $\mathcal{R}^*$ and a genetic algorithm for evolving the current subset of $\mathcal{R}^*$. The evolution of the current subset of $\mathcal{R}^*$ is implemented on a separate set of calculation nodes. For every subsequent Boolean functions, the current subset of $\mathcal{R}^*$ is used. The current subset is obtained by taking into account the results of previous runs.

We used a heterogeneous cluster based on three clusters (Cluster USU, Linux, 8 calculation nodes; umt, Linux, 256 calculation nodes; um64, Linux, 124 calculation nodes)[62].

Algorithms fgrasp and posit are used only for 3SAT. Simple genetic algorithm (SGA) and our algorithm (OA) are used for SAT and 3SAT. In the experiments, we have considered scheduling from 100 to 300 tasks. Selected experimental results are given in Tables 1–3.

Table 1 Experimental results for reduction from TRS_HCR_D to 3SAT

	Fgrasp	Posit	SGA	OA
Average time	48 min	39.3 min	1.1 h	16.4 min
Maximum time	36.4 h	32.6 h	47.2 h	19.7 h
Best time	4.1 min	4.7 min	7.3 min	17 s

Table 2 Experimental results for reduction from TRS_OCR_D to 3SAT

	Fgrasp	Posit	SGA	OA
Average time	35.2 min	33.6 min	57.2 min	15.2 min
Maximum time	27.9 h	25.4 h	35.2 h	13.3 h
Best time	2.3 min	2.4 min	1.9 min	24 s

Table 3 Experimental results for reduction from TRS_HCR_D and TRS_OCR_D to SAT

	SGA (H)	OA (H)	SGA (O)	OA (O)
Average time	59.7 min	38.5 min	43.1 min	29.4 min
Maximum time	33.9 h	21.7 h	28.3 h	18.2 h
Best time	6.2 min	46 s	37 s	13 s

In Table 3, H means TRS_HCR_D and O means TRS_OCR_D.

6 Scheduling based on particle swarm optimization

For TRS_HCR, a scheduling heuristic for dynamically scheduling workflow applications was considered in [25, 34]. This heuristic optimizes the cost of task-resource mapping based on the solution given by particle swarm optimization technique.

Pandey et al.[25] considered an optimization process such that this process uses two components: the scheduling heuristic (see Algorithm 1 in [25]), and the particle swarm optimization steps for task-resource mapping optimization (see Algorithm 2 in [25]).

Now, we consider a brief description of particle swarm optimization algorithm in [25].

$$v_i^{k+1} = \omega v_i^k + c_1 \mathrm{rand}_1 \times (\mathrm{pbest}_i - x_i^k) + c_2 \mathrm{rand}_2 \times (\mathrm{gbest} - x_i^k),$$
$$x_i^{k+1} = x_i^k + v_i^{k+1}$$

where v_i^k is the velocity of particle i at iteration k, v_i^{k+1} is the velocity of particle i at iteration $k+1$, ω is the inertia weight, c_j is the acceleration coefficients, $j = 1, 2$, rand_i is the random number between 0 and 1, $i = 1, 2$, x_i^k is the current position of particle i at iteration k, pbest_i is the best position of particle i, gbest is the position of best particle in a population, and x_i^{k+1} is the position of the particle i at iteration $k+1$.

In this algorithm, random numbers c_j and rand_i were used. We use SAT solvers as the testbed for particle swarm optimization algorithm. In particular, we consider a genetic algorithm for evolving a population of values of c_j. Also, we use a genetic algorithm for evolving a population of recurrent neural networks that predict values of rand_i. Both genetic algorithms used the optimal values of $\mathcal{MH}$ for calculating the values of fitness function. Selected experimental results for particle swarm optimization algorithm with random values of c_j and rand_i and for particle swarm optimization algorithm with evolved values of c_j and rand_i and relatively optimal values of $\mathcal{MH}$ are given in Table 4.

Table 4 Experimental results for PSO

	PSO (random)	PSO (evolved)
Average result	74%	85%
Minimum result	42%	63%
Best result	88%	97%

7 Conclusions

In this paper, we have described an approach to solve TRS_HCR and TRS_OCR problems. This approach is based on constructing logical models for these problems. Using such models, we can apply algorithms for SAT to solve TRS_HCR and TRS_OCR. Also, this model allows us to create a testbed for particle swarm optimization algorithms for scheduling workflows.

References

[1] J. Arshad, P. Townend, J. Xu. An automatic intrusion diagnosis approach for clouds. *International Journal of Automation and Computing*, vol. 8, no. 3, pp. 286–296, 2011.

[2] Y. K. Guo, L. Guo. IC cloud: Enabling compositional cloud. *International Journal of Automation and Computing*, vol. 8, no. 3, pp. 269–279, 2011.

[3] B. Li, B. Q. Cao, K. M. Wen, R. X. Li. Trustworthy assurance of service interoperation in cloud environment. *International Journal of Automation and Computing*, vol. 8, no. 3, pp. 297–308, 2011.

[4] Y. C. Liu, Y. T. Ma, H. S. Zhang, D. Y. Li, G. S. Chen. A method for trust management in cloud computing: Data coloring by cloud watermarking. *International Journal of Automation and Computing*, vol. 8, no. 3, pp. 280–285, 2011.

[5] M. Armbrust, A. Fox, R. Grifth, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia. Above the clouds: A berkeley view of cloud computing. Department of Electrical Engineering and Computer Sciences, Technical report, University of California at Berkeley, USA, 2009.

[6] R. Buyya, S. Pandey, C. Vecchiola. Cloudbus toolkit for market-oriented cloud computing. In *Proceedings of the 1st International Conference on Cloud Computing*, ACM, Berlin, Germany, pp. 24–44, 2009.

[7] S. Pandey, W. Voorsluys, M. Rahman, R. Buyya, J. E. Dobson, K. Chiu. A grid workflow environment for brain imaging analysis on distributed systems. *Concurrency and Computation: Practice & Experience*, vol. 21, no. 16, pp. 2118–2139, 2009.

[8] Amazon web services, [Online], Available: http://aws.amazon.com, February 25, 2011.

[9] GoGrid home page, [Online], Available: http://www.gogrid.com, February 25, 2011.

[10] J. D. Ullman. Np-complete scheduling problems. *Journal of Computer and System Sciences*, vol. 10, no. 3, pp. 384–393, 1975.

[11] A. Abraham, R. Buyya, B. Nath. Nature′s heuristics for scheduling jobs on computational grids. In *Proceedings of the 8th IEEE International Conference on Advanced Computing and Communications*, IEEE, Piscataway, USA, pp. 45–52, 2000.

[12] M. Aggarwal, R. D. Kent, A. Ngom. Genetic algorithm based scheduler for computational grids. In *Proceedings of the 19th Annual International Symposium on High Performance Computing Systems and Application*, IEEE, Piscataway, USA, pp. 209–215, 2005.

[13] K. Amin, G. von Laszewski, M. Hategan, N. J. Zaluzec, S. Hampton, A. Rossi. GridAnt: A client-controllable grid workflow system. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, IEEE, Piscataway, USA, pp. 5–8, 2004.

[14] T. Braun, H. Siegel, N. Beck, L. Boloni, M. Maheswaran, A. Reuther, J. Robertson, M. Theys, B. Yao, D. Hensgen, R. Freund. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. *Journal of Parallel and Distributed Computing*, vol. 61, no. 6, pp. 810–837, 2001.

[15] J. Cao, S. A. Jarvis, S. Saini, G. R. Nudd. Gridflow: Workflow management for grid computing. In *Proceedings of the 3rd International Symposium on Cluster Computing and the Grid*, IEEE, Piscataway, USA, pp. 198–205, 2003.

[16] E. Deelman, G. Singh, M. H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, D. S. Katz. Pegasus: A framework for mapping complex scientific workflows onto distributed systems. *Scientific Programming*, vol. 13, no. 3, pp. 219–237, 2005.

[17] N. Furmento, W. Lee, A. Mayer, S. Newhouse, J. Darlington. ICENI: An open grid service architecture implemented with Jini. In *Proceedings of the 2002 ACM/IEEE Conference on Supercomputing*, IEEE, Piscataway, USA, 2002.

[18] Y. Gao, H. Q. Rong, J. Z. Huang. Adaptive grid job scheduling with genetic algorithms. *Future Generation Computer Systems*, vol. 21, no. 1, pp. 151–161, 2005.

[19] S. Kim, J. B. Weissman. A genetic algorithm based approach for scheduling decomposable data grid applications. In *Proceedings of the 2004 International Conference on Parallel Processing*, IEEE, Washington, USA, vol. 1, pp. 406–413, 2004.

[20] Q. Li, Y. Guo. Optimization of resource scheduling in cloud computing. In *Proceedings of the 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, IEEE, Timisoara, Romania, pp. 315–320, 2010.

[21] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, Y. Zhao. Scientific workflow management and the kepler system: Research articles. *Concurrency and Computation: Practice & Experience*, vol. 18, no. 10, pp. 1039–1065, 2006.

[22] V. D. Martino, M. Mililotti. Sub optimal scheduling in a grid using genetic algorithms. *Parallel Computing*, vol. 30, no. 5–6, pp. 553–565, 2004.

[23] T. Oinn, M. Addis, J. Ferris, D. Marvin, M. Senger, M. Greenwood, T. Carver, K. Glover, M. R. Pocock, A. Wipat, P. Li. Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, vol. 20, no. 17, pp. 3045–3054, 2004.

[24] J. E. Orosz, S. H. Jacobson. Analysis of static simulated annealing algorithm. *Journal of Optimization Theory and Applications*, vol. 115, no. 1, pp. 165–182, 2002.

[25] S. Pandey, L. Wu, S. M. Guru, R. Buyya. A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications*, IEEE, Perth, Australia, pp. 400–407, 2010.

[26] A. Salman. Particle swarm optimization for task assignment problem. *Microprocessors and Microsystems*, vol. 26, no. 8. pp. 363–371, 2002.

[27] S. Song, Y. Kwok, K. Hwang. Security-driven heuristics and a fast genetic algorithm for trusted grid job scheduling. In *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium*, IEEE, Piscataway, USA, pp. 65–74, 2005.

[28] M. F. Tasgetiren, Y. C. Liang, M. Sevkli, G. Gencyilmaz. A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem. *European Journal of Operational Research*, vol. 177, no. 3, pp. 1930–1947, 2007.

[29] I. Taylor, I. Wang, M. Shields, S. Majithia. Distributed computing with Triana on the grid: Research articles. *Concurrency and Computation: Practice & Experience*, vol. 17, no. 9, pp. 1197–1214, 2005.

[30] E. Triki, Y. Collette, P. Siarry. A theoretical study on the behavior of simulated annealing leading to a new cooling schedule. *European Journal of Operational Research*, vol. 166, no. 1, pp. 77–92, 2005.

[31] L. Wang, H. J. Siegel, V. P. Roychowdhury, A. A. Maciejewski. Task matching and scheduling in heterogeneous computing environments using a genetic-algorithm-based approach. *Journal of Parallel and Distributed Computing*, vol. 47, no. 1, pp. 8–22, 1997.

[32] M. Wieczored, R. Prodan, T. Fahringer. Scheduling of scientific workflows in the ASKALON grid environment. *ACM SIGMOD Record*, vol. 34, no. 3, pp. 56–62, 2005.

[33] J. Yu, R. Buyya, K. Ramamohanarao. Workflow scheduling algorithms for grid computing. *Metaheuristics for Scheduling in Distributed Computing Environments*, F. Xhafa, A. Abraham, Eds., Berlin, Germany: Springer-Verlag, pp. 173–214, 2008.

[34] L. Zhang, Y. Chen, R. Sun, S. Jing, B. Yang. A task scheduling algorithm based on PSO for grid computing. *International Journal of Computational Intelligence Research*, vol. 4, no. 1, pp. 37–43, 2008.

[35] C. Zhao, S. Zhang, Q. Liu, J. Xie, J. Hu. Independent tasks scheduling based on genetic algorithm in cloud computing. In *Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing*, IEEE, Beijing, China, pp. 1–4, 2009.

[36] J. Gu, P. Purdom, J. Franco, B. Wah. Algorithms for the satisfiability (SAT) problem: A survey. *Cliques, Coloring and Satisfiability: Second DIMACS Implementation Challenge*, D. Johnson, M. Trick, Eds., Providence, USA: American Mathematical Society, pp. 19–152, 1997.

[37] C. Bessiere, E. Hebrard, T. Walsh. Local consistencies in SAT. *Theory and Applications of Satisfiability Testing*, E. Giunchiglia, A. Tacchella, Eds., Berlin, Germany: Springer-Verlag, pp. 400–407, 2003.

[38] M. Davis, G. Logemann, D. Loveland. A machine program for theorem proving. *Communications of the ACM*, vol. 5, no. 7, pp. 394–397, 1962.

[39] A. Frisch, T. J. Peugniez. Solving non-boolean satisfiability problems with stochastic local search. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, ACM, Seattle, USA, vol. 1, pp. 282–288, 2001.

[40] A. Frisch, T. Peugniez, A. Doggett, P. Nightingale. Solving non-boolean satisfiability problems with stochastic local search: A comparison of encodings. *Journal of Automated Reasoning*, vol. 35, no. 1–3, pp. 143–179, 2005.

[41] K. Iwama, S. Miyazaki. SAR-variable complexity of hard combinatorial problems. *IFIP Transactions A: Computer Science and Technology*, vol. 1, no. 1, pp. 253–258, 1994.

[42] M. Büttner, J. Rintanen. Improving parallel planning with constraints on the number of operators. In *Proceedings of the 15th International Conference on Automated Planning and Scheduling*, Monterey, USA, pp. 292–299, 2005.

[43] M. Ernst, T. Millstein, D. Weld. Automatic SAT-compilation of planning problems. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, CiteuLike, Nagoya, Japan, pp. 1169–1176, 1997.

[44] H. Kautz. SATPLAN04: Planning as satisfiability. In *Proceedings of the 4th International Planning Competition at the 14th International Conference on Automated Planning and Scheduling*, Whistler, Canada, pp. 44–45, 2004.

[45] F. Aloul, B. Al-Rawi, A. Al-Farra, B. Al-Roh. Solving employee timetabling problems using Boolean satisfiability. In *Proceedings of the IEEE Innovations in Information Technology Conference*, IEEE, Dubai, pp. 1–5, 2006.

[46] J. M. Crawford, A. B. Baker. Experimental results on the application of satisfiability algorithms to scheduling problems. In *Proceedings of the 12th National Conference on Artificial Intelligence*, ACM, Menlo Park, USA, vol. 2, pp. 1092–1097, 1994.

[47] M. A. Cruz-Chávez, R. Rivera-Lopez. A local search algorithm for a SAT representation of scheduling problems. In *Proceedings of the 2007 International Conference on Computational Science and Its Applications*, ACM, Berlin, Germany, pp. 697–709, 2007.

[48] S. O. Memik, F. Fallah. Accelerated SAT-based scheduling of control/data flow graphs. In *Proceedings of the 2002 IEEE International Conference on Computer Design: VLSI in Computers and Processors*, IEEE, Freiburg, Germany, pp. 395–400, 2002.

[49] A. Wasfy, F. Aloul. Solving the university class scheduling problem using advanced ILP techniques. In *Proceedings of the 4th IEEE GCC Conference*, IEEE, Piscataway, USA, pp. 1–5, 2007.

[50] H. H. Hoos. SAT-encodings, search space structure, and local search performance. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, ACM, Stockholm, Sweden, vol. 1, pp. 296–302, 1999.

[51] A. Gorbenko, M. Mornev, V. Popov. Planning a typical working day for indoor service robots. *IAENG International Journal of Computer Science*, vol. 38, no. 3, pp. 176–182, 2011.

[52] A. Gorbenko, M. Mornev, V. Popov, A. Sheka. The problem of sensor placement for triangulation-based localisation. *International Journal of Automation and Control*, vol. 5, no. 3, pp. 245–253, 2011.

[53] A. Gorbenko, V. Popov. On the problem of placement of visual landmarks. *Applied Mathematical Sciences*, vol. 6, no. 14, pp. 689–696, 2012.

[54] A. Gorbenko, V. Popov, A. Sheka. Localization on discrete grid graphs. *Computer, Informatics, Cybernetics and Applications*, X. He, E. Hua, Y. Lin, X. Liu, Eds., Berlin, Germany: Springer-Verlag, pp. 971–978, 2012.

[55] A toolbox for Matlab TORSCHE Scheduling, [Online], Available: http://rtime.felk.cvut.cz/scheduling-toolbox/manual/, February 26, 2012.

[56] M. Kutil, P. Sucha, R. Capek, Z. Hanzalek. Optimization and scheduling toolbox. *Matlab — Modelling, Programming and Simulations*, E. P. Leite, Ed., Rijeka, Croatia: Sciyo, pp. 239–260, 2010.

[57] J. Blazewicz, J. K. Lenstra, A. H. G. Rinnooy Kan. Scheduling subject to resource constraints: Classification and complexity. *Discrete Applied Mathematics*, vol. 5, no. 1, pp. 11–24, 1983.

[58] R. L. Graham, E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, vol. 5, no. 2, pp. 287–326, 1979.

[59] zChaff SAT solver, [Online], Available: http://www.princeton.edu/∼chaff/zchaff.html, February 26, 2012.

[60] Amazon CloudFront, [Online], Available: http://aws.amazon.com/cloudfront/, February 26, 2012.

[61] Amazon Elastic Compute Cloud (Amazon EC2), [Online], Available: http://aws.amazon.com/ec2/, February 26, 2012.

[62] Web page "Computational resources of IMM UB RAS", [Online], Available: http://parallel.imm.uran.ru/mvc_now/hardware/supercomp.htm, February 26, 2012. (In Russian)

[63] A. Gorbenko, A. Lutov, M. Mornev, V. Popov. Algebras of stepping motor programs. *Applied Mathematical Sciences*, vol. 5, no. 34, pp. 1679–1692, 2011.

[64] A. Gorbenko, V. Popov. Self-learning algorithm for visual recognition and object categorization for autonomous mobile robots. *Computer, Informatics, Cybernetics and Applications*, X. He, E. Hua, Y. Lin, X. Liu, Eds., Berlin, Germany: Springer-Verlag, pp. 1289–1295, 2012.

[65] A. Gorbenko, V. Popov, A. Sheka. Robot self-awareness: Exploration of internal states. *Applied Mathematical Sciences*, vol. 6, no. 14, pp. 675–688, 2012.

[66] A. Gorbenko, V. Popov, A. Sheka. Robot self-awareness: Temporal relation based data mining. *Engineering Letters*, vol. 19, no. 3, pp. 169–178, 2011.

[67] SATLIB — The Satisfiability Library, [Online], Available: http://people.cs.ubc.ca/∼hoos/SATLIB/index-ubc.html, February 26, 2012.

[68] F. Lardeux, F. Saubion, J. K. Hao. GASAT: A genetic local search algorithm for the satisfiability problem. *Evolutionary Computation*, vol. 14, no. 2, pp. 223–253, 2006.

Anna Gorbenko received B. Sc. degree on computer science in Department of Mathematics and Mechanics, Ural State University, Russian Federation in 2009. Currently, she is a researcher of the Department of Intelligent Systems and Robotics of Ural State University. She has (co-)authored 2 books and 17 papers, 10 conferences publications. She received Microsoft Best Paper Award from international conference SYRCoSE 2011.

Her research interests include different aspects of artificial intelligence and robotics.

E-mail: gorbenko.aa@gmail.com

Vladimir Popov received M. Sc. degree of mathematics in Department of Mathematics and Mechanics, Ural State University, Russian Federation in 1992. From 1996 to 2002, he was a Ph. D. candidate in physical and mathematical sciences in Mathematics and Mechanics Institute of Ural Branch of Russian Academy of Sciences. Since 2002, he is a professor of Ural State University. From 2006 to 2009, he was the chair of the Laboratory of Distributed Computing and Investigation of Models, Algorithms and Programs of Ural State University. Since 2009, he has been the chair of the Department of Intelligent Systems and Robotics of Ural State University. He has (co-)authored 18 books and more than 120 papers, more than 40 conferences publications. He received Microsoft Best Paper Award from international conference in 2011. In 2008, one of his paper won the Russian competitive selection of survey and analytical papers.

His research interests include different aspects of artificial intelligence and robotics.

E-mail: Vladimir.Popov@usu.ru (Corresponding author)