Applying Stochastic Integer Programming to Optimization of Resource Scheduling in Cloud Computing

Qiang Li
College of Computer Science, Sichuan University, Chengdu, China
Email: liq@scu.edu.cn

Abstract—Resource scheduling based on SLA (Service Level Agreement) in cloud computing is NP-hard problem. There is no efficient method to solve it. This paper proposes an optimal model for the problem and give an algorithm by applying stochastic integer programming technique. Applying Gröbner bases theory for solving the stochastic integer programming problem and the experimental results of the implementation are also presented.

Index Terms—SLA, Resource scheduling, Gröbner bases, Stochastic Integer Programming, Cloud Computing

I. INTRODUCTION

Cloud computing is a resource delivery and usage model to get resource (hardware, software, applications) via network "on-demand" and "at scale" as services in a multi-tenant environment. The network of providing resource is called Cloud. All resources in the cloud are scalable infinitely and used whenever as utility. In practice of cloud computing, providing an optimal/appropriate resource to user becomes more and more important.

In cloud computing, each application is often designed as a business process which includes a set of abstract services. Each abstract service encapsulates the function of an application component using its interface, and a concrete service(s) or resource(s) is selected (bound) at runtime to fulfill the function. A Service Level Agreement (SLA) in cloud computing is defined upon a business process as its end-to-end Quality of Service (QoS) constraints since a business process defines how abstract services interact to accomplish a certain business goal. Since different concrete services may operate at different QoS levels, an appropriate/optimal set of concrete services/resources may be selected so that it guarantees the fulfillment of SLA and cost is minimal. Such problem, the QoS-aware service composition problem, is a combinatorial optimization problem which ensures the optimal mapping between each abstract service and available resources [1], [2]. Since the problem is known as NP-hard [3], it takes a significant amount of time and costs to find optimal solutions (optimal combinations of resources) from a huge number of possible solutions.

This paper proposes an optimization model based on stochastic integer programming, which address the SLA-aware resource composition problem, we define a resource composition model based on stochastic integer programming and provide an algorithm to solve stochastic integer programming problem.

This paper is organized as follows. Section 2 shows the problem of resource composition and section 3 proposes a model of a resource composition based on stochastic integer programming technique. Section 4 describes the algorithm for solving stochastic integer programming based on Gröbner Bases theory [4], [5]. Section 5 presents an algorithm for solving the optimal model of resource scheduling and gives simulation results to evaluate the algorithm. Sections 6 and 7 conclude with some discussion on related work.

II. SLA-BASED RESOURCE COMPOSITION PROBLEM

We define the SLA-based resource composition problem with the following assumptions:

(1) A SLA between user and cloud provider: a service agreement on throughput, latency and cost.
(2) A business process instance: it realizes users request.
(3) A series of abstract services: it executes the business process.
(4) Concrete service/Resource: it implements an abstract service.
(5) QoS for each concrete service/resource: it has three attributes: throughput, latency and cost, while throughput and latency can vary at runtime.

The problem is how to select concrete service/resource to realize abstract service in business process instance that satisfies SLA, while the overall cost is minimal. The resource composition problem is also shown as Fig.1.

III. MODEL FOR SLA-BASED RESOURCE SCHEDULE

Applying stochastic integer programming technique, we give a model for SLA-based resource schedule. With this model, we can find the optimal resource schedule to realize a business process and satisfies SLA.

Model 3.1 Optimal Resource Scheduling Model

\[
\min \sum_{i=1}^{\alpha} \sum_{j=1}^{\beta_i} c_{ij} x_{ij}
\]
subject to:

$$\sum_{i=1}^{||\alpha||} \sum_{j=1}^{||\beta_i||} c_{ij} x_{ij} \leq C_{SLA}$$  \hspace{1cm} (2) $$

$$\sum_{j=1}^{||\beta_i||} x_{ij} = 1, i \in \{1, \cdots, ||\alpha||\}$$  \hspace{1cm} (3) $$

$$\text{Pro}\{\min\{\xi_{ij}^c x_{ij} : i \in \{1, \cdots, ||\alpha||\}, j \in \{1, \cdots, ||\beta_i||\}, x_{ij} \neq 0\} \geq T_{SLA}, \sum_{i=1}^{||\alpha||} \sum_{j=1}^{||\beta_i||} \xi_{ij}^c x_{ij} \leq L_{SLA}\} \geq \gamma$$  \hspace{1cm} (4) $$

$$x_{ij} \in \{0, 1\}$$  \hspace{1cm} (5) $$

Where

- $\alpha$: a set of Abstract services
- $|\alpha|$: number of elements in set $\alpha$
- $\beta_i$: a set of resource available to implement $i$-th abstract service
- $|\beta_i|$: number of elements in set $\beta_i$
- $x_{ij} = \{1 : \text{select } j\text{th resource for } i\text{th abstract service} \}
- \text{0 : otherwise}
- \text{ } \<c_{ij}>_{j}$: cost of $j$-th resource for implementing $i$-th abstract service
- $C_{SLA}$: cost of SLA
- $T_{SLA}$: throughput of SLA
- $L_{SLA}$: latency of SLA
- $\xi_{ij}^c$: random variable for resource’s throughput
- $\xi_{ij}^l$: random variable for resource’s latency
- $\gamma$: probability of fulfillment of a given SLA

Formula (1) means that the overall cost is minimal under the solution of variables $x_{ij}$. Formula (2) means that overall cost is less than or equal to SLA’s cost. Formula (3) means that only one resource is selected to implement an abstract service and formula (4) means that probability of fulfillment of SLA’s two attributes is great than or equal to $\gamma$.

The solution of above model is discussed in next section.

IV. GRÖBNER BASES FOR STOCHASTIC INTEGER PROGRAMMING

In this section we first introduce Gröbner Bases for integer programming (IP) and then extend it to solve stochastic integer programming (SIP).

A. Gröbner Bases for IP

Consider the following model of IP problems:

$$IP_{A,C}(b) = \min\{Cx : Ax = b, x \in \mathbb{N}^n\}$$

where $C$ is an $n$-vector of real numbers, $A$ is an $m \times n$ matrix of integers and $b$ is an $m$-vector of integers. This model means that we solve variables $x$ under the constraints $Ax = b$ so that the value of $Cx$ is minimal. We use $IP_{A,C}$ to denote a generic IP problem.

IP problems are combinatorial optimization problems where conventional numerical methods based on the hill-climbing technique can not be directly applied. Conventional methods for solving IP are based on searching algorithms where heuristics such as branch-and-bound can be applied to reduce the search space.

The method for solving IP via Gröbner Bases was firstly introduced by Conti et al. in commutative algebra [6] and by Thomas in geometry [11] independently. The key idea is to encode an IP problem into a special ideal associated with the constraint matrix $A$ and the cost (objective) function $Cx$. An important property of such an encoding is that its Gröbner bases correspond directly to the test sets of the IP problem. Thus, by employing an algebraic package such as MACAULAY or MAPLE, the test sets of the IP problem can be directly computed. Using a proper test set (such as the minimal test set which corresponds directly to the reduced Gröbner basis of the encoded ideal), the optimal value of the cost function can be computed by constructing a monotonic path from the initial non-optimal solution of the problem to the optimal solution. We can solve IP simply by following steps.

1. Encoding IP into an ideal

   This is a special ideal called toric ideal $I_A$ associated with the constraint matrix $A$ and the cost (objective) function $Cx$.

   $$I_A = \langle a^\alpha - x^\beta : \alpha, \beta \in \mathbb{N}^n, \alpha - \beta \in \text{Ker}(A) \rangle$$
results of the implementing BGBA state that MGBA shows significant performance improvement comparing to Conti [6] and Thomas [11]’s algorithms.

B. Apply MGBA to Solve SIP

Stochastic programming with probabilistic constraints as a decision model under uncertainty is called chance constrained programming. If some or all variables in the chance constrained programming problems are integers, it is called chance constrained IP (CIP).

We abstract the following model as a general class of CIP:

$$\min h(x) \text{ subject to } \frac{\text{Prob}(Tx \geq \xi)}{A} \geq \gamma,$$

where $x \in \mathbb{N}^n$ and $\xi$ is a vector of random variables.

This model has two properties: (1) the probabilistic constraint is not separable and (2) integer variables are required to model setup decision. The traditional techniques for chance constrained programming do not provide a satisfactory solution for models that have integer variables and nonseparable chance (probabilistic) constraint arising from nonnormal distributions.

Based on S.R.Tayur et al.’s idea [10], we can solve this problem by applying MGBA. The idea is dividing the problem into two parts: one is composed of the probabilistic constraint and some complicated constraints, called membership oracle and the other is a simple IP after removing the membership oracle from the problem, called reduced IP. We first compute the test set (reduced Gröbner basis of a toric ideal) for reduced IP by using MGBA. The test set provides a set of directions that can be used to trace paths from every nonoptimal solution to the optimal solution of the reduced IP. So, for any feasible solution of the reduced IP, we get an optimal solution by searching the set of directions. Simultaneously, we can also walk back from the optimal solution to every feasible solution of the reduced IP by simply reversing these paths. So, with the same test set, we find the optimal solution of the CIP by walking back from the optimal solution of the reduced IP to other feasible solutions and querying the membership oracle to check whether the reached point is feasible for the CIP. We prove that the search terminates with either the optimal solution of CIP or all paths are searched, i.e., the CIP is infeasible.

To introduce the algorithm for solving CIP, we first discuss the test set for CIP in detail.

We consider a class of the chance constrained IP problems of the form:

CIP: Min $ce$ subject to $Ax = b$ \quad $\frac{\text{Prob}(Tx \geq \xi)}{A} \geq \gamma$ \quad $x \in \mathbb{N}^n$

where $\xi$ is a vector of random variables, $c \in \mathbb{R}^n$, $A \in \mathbb{Z}^{m \times n}$, $T \in \mathbb{R}^{m \times p}$, $b \in \mathbb{Z}^m$ and $\gamma \in \mathbb{R}$. First we divide...
CIP into two parts: reduced IP (RIP) and membership oracle. The reduced IP is the form as follow.

RIP: \[
\text{Min } cx \text{ subject to } \\
Ax = b \\
x \in \mathbb{N}^n
\]
The membership oracle is composed of \(Prob\{Tx \geq \xi\} \geq \gamma\). In some models for practical problems, the membership oracle may includes some complicated constraints except probability constraints.

By MGBA, we compute a test set for RIP and derive an optimal solution of RIP by using this test set to reduce any feasible solution of RIP. Now we need a test set for CIP, which provides a set of directions that can be used to walk in a systematic manner from the optimal solution of RIP to other solutions, querying the membership oracle each time to check feasibility with respect to CIP. The walking always terminates at the optimum for CIP or finding CIP infeasible. First we give the definition of test set for CIP.

**Definition 4.1** A set \(\Omega \subseteq \mathbb{Z}^n\) is a test set for CIP (with respect to \(\alpha\)) if it provides a set of directions so that

1. walking from RIP optimum using the directions always lands on a point that is feasible for RIP if nonnegativity constraints are satisfied;
2. all feasible points of CIP can be reached using only the directions;
3. on every path, every step deeper into the polytope (starting from the RIP optimum) increases the cost monotonely.

Let \(G_{\alpha}\) be the minimal \(\{\alpha(i) - \beta(i), i = 1, \ldots, s\}\) be the test set for RIP. We construct a set \(\Omega\) for CIP as follows. Let \(\Omega = \{\beta(i) - \alpha(i), i = 1, \ldots, s\}\) i.e., reverse all directions in the test set \(G_{\alpha}\). We have the following theorem.

**Theorem 4.1** The set \(\Omega\) is a test set for CIP.

**Proof:** Consider the \(\alpha\)-skeleton of the appropriate fiber. Suppose we now reverse the directions of all edges in this skeleton and call the resulting graph the reverse \(\alpha\)-skeleton of the fiber. By Corollary in [9], in the \(\alpha\)-skeleton of a fiber, there exists a directed path from every nonoptimal point \(\alpha\) of RIP to the unique optimum \(\beta\). Thus, for any feasible point \(\alpha\) of RIP, there exists a directed path \(P(\alpha)\) in the reverse \(\alpha\)-skeleton of this fiber, from the RIP optimum to this feasible in the fiber. Because the set of all feasible points of CIP is a subset of one of RIP, all feasible points of CIP can be reached by the reverse \(\alpha\)-skeleton. Also, the objective function value of points reached as the path is traversed from the RIP optimum to \(\alpha\), increases monotonically.

So, we only show that we can build the reverse \(\alpha\)-skeleton in a fiber by using \(\Omega\). Let \(\Omega\) denote the set of vectors in \(G_{\alpha}\) with directions reversed. At a feasible lattice point \(\theta\) in a fiber of RIP, we draw all vectors from \(\Omega\) that can be translated through some \(v \in \mathbb{N}^n\) such that its tail is incident at \(\theta\). As before, the heads of the translated vectors are also incident at feasible lattice points in the same fiber. Since only the vectors in \(G_{\alpha}\) were reversed, this construction and the previous one have the same underlying undirected graph in every fiber. This proves the claim.

The definition and theorem about the test set for CIP provide us an algorithm for finding the optimal solution of CIP from the RIP optimum. We use the paths in the reverse \(\alpha\)-skeleton of RIP to find the CIP optimum \(X_0\). Let \(\beta\) be the RIP optimum and \(P(\alpha)\) be a path in the reverse \(\alpha\)-skeleton of the fiber from \(\beta\) to \(\alpha\). Let \(\Omega\) be a set of path in the form \(P(\alpha)\). We may assume that \(\beta\) is infeasible for CIP since otherwise we have done.

**Algorithm 4.1 Feasible Checking Algorithm**

This algorithm checks whether CIP is feasible or infeasible, if feasible, its optimum will be found.

**Initialize:** \(\Omega = \{P(\beta)\}\), \(X_0 = M (M \in \mathbb{N}^n\) is a vector whose components have large magnitude.)

**Repeat** For \(P(\alpha) \in \Omega\), let \(P(\omega_1), \ldots, P(\omega_q)\) be the paths obtained by adding to \(P(\alpha)\) those edges in \(\Omega\) such that \(\omega_i \geq 0\). This ensures that \(P(\omega_i)\) is a path in the reverse \(\alpha\)-skeleton. A path that leads to an infeasible point may be assumed to be pruned at \(\alpha\).

For \(i = 1\) to \(q\)

if \(\omega_i\) feasible for CIP and \(X_0 \geq \omega_i\), let \(X_0 = \omega_i\) and prune \(P(\omega_i)\).

else if \(\omega_i \geq \alpha\) \(X_0\) then prune \(P(\omega_i)\)

else \(\Omega = \Omega \cup \{P(\omega_i)\}\)

**Until** all paths in \(\Omega\) are pruned.

**Theorem 4.2** Algorithm 4.1 terminates in finite steps and

1. CIP is infeasible if and only if \(X_0 = M\), or
2. CIP is feasible, \(X_0\) is an optimal solution of CIP.

**Proof:** Finiteness of the algorithm is clear since the optimal solution of CIP is bounded with respect to \(\alpha\).

Also, it is clear that CIP is infeasible if and only if \(X_0 = M\). Now we show \(X_0\) is the CIP optimum when CIP is feasible.

Let \(\mu\) be any feasible of CIP. Clearly \(\mu\) is feasible for RIP and there exists a path \(P(\mu)\), in the reverse \(\alpha\)-skeleton, from the optimum \(\beta\) of RIP to \(\mu\). Note that \(P(\mu)\) may not be unique but any such path will suffice for this proof. We identify \(P(\mu)\) with the sequence of nodes in the reverse \(\alpha\)-skeleton that constitute it, i.e., \(P(\mu) = \{\beta = u_1, \ldots, u_m = \mu\}\). Now let \(j \in \{1, \ldots, m\}\) be the largest number such that \(P(u_j)\) is in the set \(\Omega\) of the above procedure at some stage. We have the following two cases:

(i) \(P(u_j)\) was pruned since \(u_j\) became feasible for CIP.

In this case, \(u_j \geq X_0\), that is \(c u_j \geq c X_0\). Also \(c\mu \geq c u_j\), since the successive nodes in \(P(\mu)\) have monotonically increasing cost when the path is traversed from \(\beta\) to \(\mu\). Therefore \(c\mu \geq c X_0\).
(ii) $P(u_j)$ was pruned since $u_j > cX_0$. Again $c\mu \geq cu_j \geq cX_0$. Therefore, we have shown that in either case, we have $c\mu \geq cX_0$. So, $X_0$ is an optimal solution when CIP is feasible.

Combining the feasible checking algorithm and MGBA, we can provide an algorithm for solving CIP. Before describing this algorithm, we first consider another part of the CIP, membership oracle.

The membership oracle is composed of the probabilistic constraint. Some times in order to keep the reduced IP as simple as possible, the membership oracle also includes some complicated constraints. In each step of walking the directions of the test set $\Omega$ from the RIP optimum to a feasible solution, we need to query the membership oracle, i.e., check whether the point is feasible for the CIP. Assume we have a probabilistic constraint $\text{Prob}(Tx \geq \xi) \geq \gamma$ in the membership oracle and a set of samples $\{\xi_i, i = 1, \ldots, s\}$. Then a feasible point $v$ of RIP is feasible for CIP if and only if

$$Tv \leq \xi_i, \; i \in \Xi \subseteq \{1, \ldots, s\} \text{ and } |\Xi| \geq \gamma s$$

Now we give a whole algorithm for chance constrained IP.

**Algorithm 4.2 Algorithm for CIP**

1) Decompose CIP into RIP and membership oracle.
2) Compute the test set $G_\triangleright$ of RIP by using MGBA.
3) Compute an optimal solution $x^R$ from any feasible solution of RIP.
4) Derive the test set $\Omega$ of CIP by reversing all directions of vectors in $G_\triangleright$.
5) Compute the optimal solution of CIP with the test set $\Omega$ and the RIP optimum $x^R$ by using the feasible checking algorithm (Algorithm 4.1).

We give an example to illustrate the method.

\[
\begin{align*}
\text{min} & \quad 100y + x_1 + x_2 \\
\text{subject to} & \quad y + 3x_1 + 2x_2 = 6 \\
& \quad \text{Prob}\{100y + x_1 + x_2 \geq \xi_1, (100y + x_1 + x_2 \leq \xi_2)\} \geq \gamma \\
& \quad y, x_1, x_2 \in \mathbb{N}
\end{align*}
\]

Here $\gamma = 0.8$,

$$\xi = (\xi_1, \xi_2) = \{ (202, 215) \; 80\%, (106, 115) \; 20\% \}$$

Firstly we compute the test set of the following RIP:

\[
\begin{align*}
\text{min} & \quad 100y + x_1 + x_2 \\
\text{subject to} & \quad y + 3x_1 + 2x_2 = 6
\end{align*}
\]

We get the test set as follow.

\[
\begin{align*}
g_1 &= [(2, 0, 0), (0, 0, 1)], \quad g_2 = [(1, 0, 1), (0, 1, 0)], \\
g_3 &= [(1, 1, 0), (0, 0, 2)], \quad g_4 = [(0, 0, 3), (0, 2, 0)].
\end{align*}
\]

Then by reversing all directions of above test set, we get the test set of SIP as follow.

\[
\begin{align*}
\tilde{g}_1 &= [(0, 0, 1), (2, 0, 0)], \quad \tilde{g}_2 = [(0, 1, 0), (1, 0, 1)], \\
\tilde{g}_3 &= [(0, 0, 2), (1, 1, 0)], \quad \tilde{g}_4 = [(0, 2, 0), (0, 0, 3)].
\end{align*}
\]

Deriving the optimal solution from the feasible solution $(0,2,0)$ by using the test set, we get the optimal solution: $(2,0,2)$, as Fig.3.

**V. OPTIMAL ALGORITHM AND IMPLEMENTATION**

An algorithm for solving the optimal resource scheduling model in cloud computing is given as follow.

**Algorithm 5.1 Algorithm for Model 3.1**

1) Decompose Model 3.1 into RIP (including formula (1),(2),(3) and (5)) and membership oracle (including formula (4)).
2) Compute a test set $G$ of RIP by using MGBA
3) Compute an optimal solution from any feasible solution of RIP
4) Derive the test set $F$ of Model3.1 by reversing all directions of vectors in $G$
5) Compute the optimal solution of Model3.1 with test set $F$ and optimum of RIP by using feasible checking method

We have implemented MGBA for SIP and have done several simulation by finding optimal resource schedule for business process with different abstract services and the different SLA. The simulation of resource scheduling optimization includes the following case studies:
(1) A user’s process includes 2 abstract services. Each abstract service has 3 resources available to be selected, while the probability of fulfillment of SLA is 0.84 or 0.88.

(2) A user’s process includes 4 abstract services. Each abstract service has 3 resources available to be selected, while the probability of fulfillment of SLA is 0.5.

(3) A user’s process includes 6 abstract services. Each abstract service has 3 resources available to be selected, while the others have 6 resources available to be selected. The probability of fulfillment of SLA is 0.62.

(4) A user’s process includes 7 abstract services. One abstract service has 5 resources available to be selected, while the others have 6 resources available to be selected. The probability of fulfillment of SLA is 0.47.

The experiment result is shown in Table 1.

From the result, we can see that the computation of optimal resource schedule finished in a reasonably short time. But the time is growing exponentially with the increasing of the numbers of the services and resources. The reason is that the computation of Gröbner basis still suffers from the fact that the size of Gröbner basis grows exponentially with the increasing of the variables.

VI. RELATED WORKS

In [13], H.Wada et al. develop a multi-objective optimization model to tackle SLA-aware service composition problem. They consider multiple SLAs simultaneously and provided a set of solutions of equivalent quality. But their model is based on the heuristic genetic algorithm in which good/acceptable performance can not be expected.

In [14], S.Chaisiri et al. apply stochastic integer programming for resource provision optimization problem. The algorithm minimizes the total cost of resource provision in a cloud computing environment. The optimal solution is obtained by formulating and solving stochastic integer programming with two-stage recourse. However, they do not consider the notion of SLA, which is one of the most important business notion in cloud computing.

VII. CONCLUSION

In this paper, we have proposed a model for optimization of SLA-based resource schedule in cloud computing based on stochastic integer programming technique. By applying Gröbner bases theory, we extend MGBA to solve the stochastic integer programming with two-stage recourse, furthermore introduce a method to solve the optimal model of SLA-based resource schedule problem. The performance evaluation has been performed by numerical studies and simulation. The experimental result shows that the optimal solution is obtained in a reasonable short time.

We plan to have several extensions as future work. The resource scheduling optimization model will be extended to tackle multi-objective optimization problems. The MGBA for stochastic integer programming will be extended to solve multi-objective stochastic integer programming problems. The overall performance of the algorithm will be improved by improving the computation of Gröbner Basis/test set for integer programming.

REFERENCES


Qiang Li is an associate professor at the college of computer science, Sichuan university in Chengdu, China. His research fields are symbolic computing, cloud computing, internet of things and mobile computing. In March 1990, Qiang obtained his PhD at the department of computer science, Tsukuba university in Japan and after that he worked for software research/development at Fujitsu, Nokia in Tokyo, and Access company separately. In April 2010, he came back to china and joined to the college of computer science, Sichuan university.