

## A Stochastic Knapsack Problem with Continuous Random Capacity

Suwitchporn Witchakul, Prapaisri Sudas Na Ayudhaya and Peerayuth Charnsethikul  
Department of Industrial Engineering, Operations Research and Management Science Units,  
Kasetsart University, Bangkok 10903, Thailand

---

**Abstract: Problem Statement:** The problem of allocating a set of items in order to maximize the total linear profit under uncertain capacity referred as a stochastic knapsack problem with continuous random capacity was studied theoretically and computationally. **Approach:** Two optimization based heuristic algorithms were proposed and developed for solving the problem and both efficiency and effectiveness are compared with the general purpose method using the Monte Carlo simulation. **Results:** For both the relaxed and the original problems, both algorithms with appropriate stepping size parameter were superior on average to the simulation approach in both computing time and solution quality. **Conclusion:** Therefore, both proposed approaches can be practical for large scale problem and can be used as a basic algorithm for a more complex nature in case of simultaneously random in both items weight and knapsack capacity.

**Key words:** Stochastic knapsack problems, decision under uncertainties, infinite programming

---

### INTRODUCTION

A common integer linear programming (ILP) used to decide which subset of  $n$  items or projects should be selected such that the total profit is maximized without exceeding a given capacity or budget is called the Knapsack Problem (KP). The knapsack problem has been extensively studied in operations research<sup>[1]</sup>. The knapsack problem has various industrial applications such as resource allocation problem, capital budgeting problem, project selection problem, cargo loading problem, cutting stock problem and many others. The knapsack problem of this type is deterministic because all parameters are known with certainty. However, in realistic situations, these parameters may not be known with certainty. Therefore, they have to be considered as random variables having a certain distribution. Thus, this type of problem is called the stochastic knapsack problem. There are few researchers concerning with solving the stochastic knapsack problem, which are usually concentrated with special cases, e.g., Parks and Steinberg<sup>[2]</sup>, Sniedovich<sup>[3]</sup> and Henig<sup>[4]</sup> studied static stochastic knapsack problem when benefit is uncertain. Kleywegt and Papastavrou<sup>[5]</sup> studied dynamic stochastic knapsack problem when benefit is uncertain. Chiu *et al.*<sup>[6]</sup>, Kleywegt *et al.*<sup>[7]</sup>, Slyke and Young<sup>[8]</sup> and Kleywegt and Papastavrou<sup>[9]</sup> studied dynamic stochastic knapsack problem when benefit and weight are uncertain.

All of the above works concern with studying uncertainty in weight parameters and/or benefit

parameters. However, studying uncertainty in capacity is also important. For example, in capital budgeting problem, sometimes, the budget is uncertainty (e.g., there are  $n$  alternatives of budget with the probability for each alternative or the budget is random with known probability distribution). In resource allocation problem, sometimes, the available resource is uncertainty (e.g., the available resource is random with known probability distribution). In cargo loading problem, when the truck has many customers to load the items. The available space is unknown before the truck arrived. Hence, the remaining capacity of the truck is uncertainty. There are two types of uncertainty. One possibility is to estimate the probability of occurrence of each state of nature (i.e., there are the previous information). Another is not possible to estimate the probabilities of the states of nature. In this research, the stochastic knapsack problem with continuous probability distribution on capacity was studied.

The studied problem is the problem in which the allocation of decision variable  $x_j$  is made to meet random capacity with a known distribution. The objective function in this problem may be a total cost, which can be divided into two parts. The first part is the cost of selecting the decision variables  $x_j$  and the second part is the penalty cost (i.e., shortage and storage cost). When the capacity has continuous probability distribution, the problem is called Stochastic Knapsack Problem with Continuous Random Capacity (SKPCRC). It can be stated as follows:

---

**Corresponding Author:** Peerayuth Charnsethikul, Department of Industrial Engineering, Operations Research and Management Science Units, Kasetsart University, Bangkok 10903, Thailand

Minimize  $f =$

$$\sum_{j=1}^n c_j x_j + g \int_L^U p(b)u(b)db + h \int_L^U p(b)v(b)db$$

Subject to  $\sum_{j=1}^n a_j x_j + u(b) - v(b) = b$

$0 \leq x_j \leq t_j$  and integer

$$\int_L^U p(b)db = 1$$

where  $x_j$  is the decision variable, for  $j = 1, 2, \dots, n$ ,  $t_j$  is the upper bound of  $x_j$ , for  $j = 1, 2, \dots, n$ ,  $u(b)$  is the slack variable and  $u(b) \geq 0$ , for  $L \leq b \leq U$ ,  $v(b)$  is the surplus variable and  $v(b) \geq 0$ , for  $L \leq b \leq U$ ,  $a_j$  is the weight coefficient of item  $j$  and  $a_j \geq 0$ , for  $j = 1, 2, \dots, n$ ,  $c_j$  is the cost coefficient of item  $j$  and  $c_j \geq 0$ , for  $j = 1, 2, \dots, n$ ,  $b$  is the capacity,  $p(b)$  is the probability of having capacity  $b$ ,  $p(b) \geq 0$ , for  $L \leq b \leq U$ ,  $g$  is the per unit cost of having  $u(b)$ ,  $g \geq 0$  and  $h$  is the per unit cost of having  $v(b)$ ,  $h \geq 0$ .

### MATERIALS AND METHODS

#### Materials:

- A personal computer, Intel Pentium M Processor 1.6 GHz., 512 MB RAM, 60 GB hard drive
- MATLAB 7.0.1 software
- CPLEX 8.1 software (ILOG<sup>[10]</sup>)
- Visual C.NET software

**Methods:** According to the model, if integer constraint is relaxed, then the relaxed problem is SKPCRCR that can be stated as follows. Similarly to the discrete case, two algorithms were proposed for solving SKPCRCR. The following assumption was assumed.

$$c_j / a_j \leq c_{j+1} / a_{j+1}, \text{ for } j = 1, 2, \dots, n - 1$$

**Algorithm 1 of SKPCRCR:** This algorithm is based on the original approach of Witchakul and Sudas-na-Ayudhaya<sup>[11]</sup>. There exist three optimality conditions adapted from<sup>[12]</sup> that are following:

**Case 1:** The optimal solution is as follows:

$$\begin{aligned} x_j &= 0, \forall j \\ v(b) &= 0, L \leq b \leq U \\ u(b) &= b, L \leq b \leq U \\ \text{where } c_1 / a_1 &\geq g \end{aligned}$$

**Proof:**  $u(b) = b - \sum_{j=1}^n a_j x_j + v(b), L \leq b \leq U$

Next part is to substitute these basic variables into the objective function as follows:

$$\begin{aligned} f &= \sum_{j=1}^n c_j x_j + g \int_L^U p(b)(b - \sum_{j=1}^n a_j x_j + v(b))db + h \int_L^U p(b)v(b)db \\ &= g \int_L^U p(b)bdb + (g + h) \int_L^U p(b)v(b)db + \sum_{j=1}^n (c_j - ga_j) \int_L^U p(b)db x_j \\ &= g \int_L^U p(b)bdb + (g + h) \int_L^U p(b)v(b)db + \sum_{j=1}^n (c_j - ga_j)x_j \end{aligned}$$

Since, the minimum  $f$  can be found when all reduced costs of nonbasic variables are greater than or equal to zero. In this case, nonbasic variables are  $x_j$ , for all  $j$  and  $v(b)$  for all  $b$ . In order to obtain the minimum value of  $f$ , the reduced costs of these nonbasic variables must be greater than or equal to zero that are  $(g + h) \int_L^U p(b)v(b)db$  for all  $b$  and  $(c_j - ga_j) \geq 0$ , for all  $j$ . Since  $g, h \geq 0$  and  $p(b) \geq 0$ , for  $L \leq b \leq U$ ,  $(g + h) \int_L^U p(b)v(b)db$  is greater than or equal to zero. To minimize  $f$ ,  $(c_j - ga_j) \geq 0$ , for all  $j$ , that is equivalent to  $c_j / a_j \geq g$ , for all  $j$ . According to the assumption  $c_j / a_j \leq c_{j+1} / a_{j+1}$ , for  $j = 1, 2, \dots, n - 1$ , the required condition for this case is  $c_1 / a_1 \geq g$ . In this case, the minimum value of  $f$  is  $g \int_L^U p(b)bdb$ .

**Case 2:** The optimal solution is as follows:

$$x_j = t_j, \forall j$$

$$v(b) = \sum_{j=1}^n a_j t_j - b \text{ and } u(b) = 0, \quad b \in B_2$$

$$u(b) = b - \sum_{j=1}^n a_j t_j \text{ and } v(b) = 0, \quad b \in B_1$$

where

$$B_1 = \{L \leq b \leq U : b \geq \sum_{j=1}^n a_j t_j\},$$

$$B_2 = \{L \leq b \leq U : b < \sum_{j=1}^n a_j t_j\} \text{ and}$$

$$c_n / a_n \leq g \int_{b \in B_1} p(b)db - h \int_{b \in B_2} p(b)db.$$

**Proof:**  $x_j = t_j - r_j, \forall j$

$$r_j \geq 0, \forall j$$

$$u(b) = b - \sum_{j=1}^n a_j t_j + \sum_{j=1}^n a_j r_j + v(b), \quad b \in B_1$$

$$v(b) = -b + \sum_{j=1}^n a_j t_j - \sum_{j=1}^n a_j r_j + u(b), \quad b \in B_2$$

Next part is to substitute these basic variables into the objective function as follows:

$$\begin{aligned} f &= \sum_{j=1}^n c_j (t_j - r_j) + g \int_{b \in B_1} p(b) (b - \sum_{j=1}^n a_j t_j + \sum_{j=1}^n a_j r_j + v(b)) db \\ &\quad + h \int_{b \in B_1} p(b) v(b) db + h \int_{b \in B_2} p(b) (-b + \sum_{j=1}^n a_j t_j - \sum_{j=1}^n a_j r_j + u(b)) db \\ &\quad + g \int_{b \in B_2} p(b) u(b) db \\ &= \sum_{j=1}^n c_j t_j + g \int_{b \in B_1} p(b) (b - \sum_{j=1}^n a_j t_j) db + h \int_{b \in B_2} p(b) (-b + \sum_{j=1}^n a_j t_j) db \\ &\quad + (g+h) \int_{b \in B_1} p(b) v(b) db + (g+h) \int_{b \in B_2} p(b) u(b) db \\ &\quad + \sum_{j=1}^n (-c_j + g a_j \int_{b \in B_1} p(b) db - h a_j \int_{b \in B_2} p(b) db) r_j \end{aligned}$$

In this case, nonbasic variables are  $v(b)$ , for  $b \in B_1$ ,  $u(b)$ , for  $b \in B_2$  and  $r_j$ , for all  $j$ . In order to obtain the minimum value of  $f$ , the reduced costs of these nonbasic variables must be greater than or equal to zero that are  $(g+h) \int_{b \in B_1} p(b) v(b) db \geq 0$ ,  $(g+h) \int_{b \in B_2} p(b) v(b) db \geq 0$  and  $(-c_j + g a_j \int_{b \in B_1} p(b) db - h a_j \int_{b \in B_2} p(b) db)$ , for all  $j$ .

Since  $g, h \geq 0$  and  $p(b) \geq 0$ , for  $L \leq b \leq U$ ,  $(g+h) \int_{b \in B_1} p(b) v(b) db$  and  $(g+h) \int_{b \in B_2} p(b) v(b) db$  are greater than or equal to zero. To minimize  $f$ ,  $(-c_j + g a_j \int_{b \in B_1} p(b) db - h a_j \int_{b \in B_2} p(b) db)$ , for all  $j$  must be greater than or equal to zero that is equivalent to  $c_j / a_j \leq g \int_{b \in B_1} p(b) db - h \int_{b \in B_2} p(b) db$ , for all  $j$ . According to the assumption  $c_j / a_j \leq c_{j+1} / a_{j+1}$ , for  $j=1, 2, \dots, n-1$ , the condition for this case is  $c_n / a_n \leq g \int_{b \in B_1} p(b) db - h \int_{b \in B_2} p(b) db$ . In this case, the

minimum value of  $f$  is  $\sum_{j=1}^n c_j t_j + g \int_{b \in B_1} p(b) (b - \sum_{j=1}^n a_j t_j) db + h \int_{b \in B_2} p(b) (-b + \sum_{j=1}^n a_j t_j) db$ .

**Case 3:** The optimal solution is as follows:

$$x_j = t_j, \quad j=1, \dots, k-1$$

$$x_k = \frac{B - \sum_{j=1}^{k-1} a_j t_j}{a_k}$$

$$x_j = 0, \quad j = k+1, \dots, n$$

$$v(b) = B - b \text{ and } u(b) = 0, \quad L \leq b < B$$

$$u(b) = 0 \text{ and } v(b) = 0, \quad b = B$$

$$u(b) = b - B \text{ and } v(b) = 0, \quad B < b \leq U$$

where

$B \in [L, U]$  and  $k \in \{1, \dots, n\}$  such that

$$(c_k / a_k + h \int_L^B p(b) db - g \int_B^U p(b) db) = 0,$$

$$\sum_{j=1}^{k-1} a_j t_j \leq B, \quad \sum_{j=1}^k a_j t_j \geq B. \text{ If } \sum_{j=1}^n a_j t_j < B, \text{ then } k = n.$$

**Proof:** Suppose  $k$  and  $B$  have been specified.

$$\begin{aligned} f(B) &= \sum_{j=1}^{k-1} c_j x_j + c_k x_k + \sum_{j=k+1}^n c_j x_j + g \int_L^B p(b) u(b) db + h \int_L^B p(b) u(b) db \\ &\quad + g \int_B^U p(b) u(b) db + h \int_B^U p(b) u(b) db \end{aligned} \quad \text{Let,}$$

$$x_j = t_j, \quad j=1, \dots, k-1,$$

$$x_k = (B - u(B) + v(B) - \sum_{j=1}^{k-1} a_j x_j - \sum_{j=k+1}^n a_j x_j) / a_k$$

$$= (B - u(B) + v(B) - \sum_{j=1}^{k-1} a_j t_j - \sum_{j=k+1}^n a_j x_j) / a_k,$$

$$v(b) = \sum_{j=1}^{k-1} a_j x_j + a_k x_k + \sum_{j=k+1}^n a_j x_j + u(b) - b$$

$$= \sum_{j=1}^{k-1} a_j x_j + a_k ((B - u(B) + v(B) - \sum_{j=1}^{k-1} a_j x_j - \sum_{j=k+1}^n a_j x_j) / a_k)$$

$$+ \sum_{j=k+1}^n a_j x_j + u(b) - b$$

$$= B - b + u(b) - u(B) + v(B), \quad L \leq b < B$$

and

$$u(b) = b + v(b) - \sum_{j=1}^{k-1} a_j x_j - a_k x_k - \sum_{j=k+1}^n a_j x_j$$

$$\begin{aligned}
 &= b + v(b) - \sum_{j=1}^{k-1} a_j x_j - a_k ((B - u(B) + v(B) - \sum_{j=1}^{k-1} a_j x_j - \sum_{j=k+1}^n a_j x_j) / a_k) \\
 &\quad - \sum_{j=k+1}^n a_j x_j \\
 &= b - B + v(b) + u(B) - v(B), \quad B < b \leq U.
 \end{aligned}$$

The next part is to substitute these basic variables into  $f(B)$ . Thus, the function becomes as follows:

$$\begin{aligned}
 f(B) &= \sum_{j=1}^{k-1} c_j t_j + c_k ((B - u(B) + v(B) - \sum_{j=1}^{k-1} a_j t_j - \sum_{j=k+1}^n a_j x_j) / a_k) + \sum_{j=k+1}^n c_j x_j \\
 &\quad + g \int_L^B p(b)u(b)db + h \int_L^B p(b)(B - b + u(b) - u(B) + v(B))db \\
 &\quad + g \int_B^U p(b)(b - B + v(b) + u(B) - v(B))db + h \int_B^U p(b)v(b)db \\
 &= \sum_{j=1}^{k-1} c_j t_j + c_k ((B - \sum_{j=1}^{k-1} a_j t_j) / a_k) + h \int_L^B p(b)(B - b)db \\
 &\quad + g \int_B^U p(b)(b - B)db + \sum_{j=k+1}^n (c_j - c_k a_j / a_k) x_j \\
 &\quad + (g + h) \int_L^B p(b)u(b)db + (g + h) \int_B^U p(b)v(b)db \\
 &\quad + (-c_k / a_k + g \int_B^U p(b)db - h \int_L^B p(b)db)u(B) \\
 &\quad + (c_k / a_k + h \int_L^B p(b)db - g \int_B^U p(b)db)v(B)
 \end{aligned}$$

In this case, nonbasic variables are  $x_j$ , for  $j = k + 1, k + 2, \dots, n$ ,  $u(b)$ , for  $L \leq b < B$ ,  $v(b)$ , for  $B < b \leq U$ ,  $u(B)$  and  $v(B)$ . In order to obtain the minimum value of  $f$ , the reduced costs of these nonbasic variables must be greater than or equal to zero that are  $c_j - c_k a_j / a_k \geq 0$ , for  $j = k + 1, k + 2, \dots, n$ ,  $(g + h) \int_L^B p(b)u(b)db \geq 0$ ,  $(g + h) \int_B^U p(b)v(b)db \geq 0$ ,  $(-c_k / a_k + g \int_B^U p(b)db - h \int_L^B p(b)db) \geq 0$  and  $(c_k / a_k + h \int_L^B p(b)db - g \int_B^U p(b)db) \geq 0$ .

From the above assumption,  $c_j / a_j \leq c_{j+1} / a_{j+1}$ , for  $j = 1, 2, \dots, n - 1$  then  $c_j - c_k a_j / a_k \geq 0$ , for  $j = k + 1, k + 2, \dots, n$ . Since  $g, h \geq 0$  and  $p(b) \geq 0$ , for  $L \leq b \leq U$ , to minimize  $f$ ,  $(-c_k / a_k + g \int_B^U p(b)db - h \int_L^B p(b)db)$  and  $(c_k / a_k + h \int_L^B p(b)db - g \int_B^U p(b)db)$  must be greater than or equal to zero that is equivalent to  $(c_k / a_k + h \int_L^B p(b)db - g \int_B^U p(b)db) = 0$ . Since

$0 \leq x_k \leq t_k$ , it is equivalent to  $0 \leq \frac{B - \sum_{j=1}^{k-1} a_j t_j}{a_k} \leq t_k$ . It is also equivalent to  $\sum_{j=1}^{k-1} a_j t_j \leq B$  and  $\sum_{j=1}^k a_j t_j \geq B$ . If  $\sum_{j=1}^n a_j t_j < B$ , then  $k = n$ .

Therefore, we select  $B$  and  $k$  such that  $(c_k / a_k + h \int_L^B p(b)db - g \int_B^U p(b)db) = 0$ ,  $\sum_{j=1}^{k-1} a_j t_j \leq B$  and  $\sum_{j=1}^k a_j t_j \geq B$ . If  $\sum_{j=1}^n a_j t_j < B$ , then  $k = n$ . In this case, the minimum value of  $f$  is  $\sum_{j=1}^{k-1} c_j t_j + c_k ((B - \sum_{j=1}^{k-1} a_j t_j) / a_k) + h \int_L^B p(b)(B - b)db + g \int_B^U p(b)(b - B)db$ .

**Algorithm 2 of SKPCRCR:** his algorithm was used to find the optimal solution of the SKPCRCR by relaxing the original problem model and utilizing the approach from [13] as follows:

$$\text{Minimize } f = \sum_{j=1}^n c'_j y_j + g \int_L^U p(b)u(b)db + h \int_L^U p(b)v(b)db$$

$$\text{Subject to } \sum_{j=1}^n y_j + u(b) - v(b) = b$$

$$0 \leq y_j \leq t'_j$$

$$\int_L^U p(b)db = 1$$

where  $y_j = a_j x_j$ ,

$$c'_j = \frac{c_j}{a_j}, \text{ and}$$

$$t'_j = a_j t_j.$$

Objective function (6.1) can be separated into two parts as follows:

$$\text{Minimize } f = H_1(\theta) + H_2(\theta)$$

where:

$$H_1(\theta) = \sum_{j=1}^n c'_j y_j$$

$$H_2(\theta) = g \int_L^U p(b)u(b)db + h \int_L^U p(b)v(b)db$$

Graphs of  $H_1(\theta)$  and  $H_2(\theta)$  of SKPCRCR were shown in Fig. 1 and 2, respectively

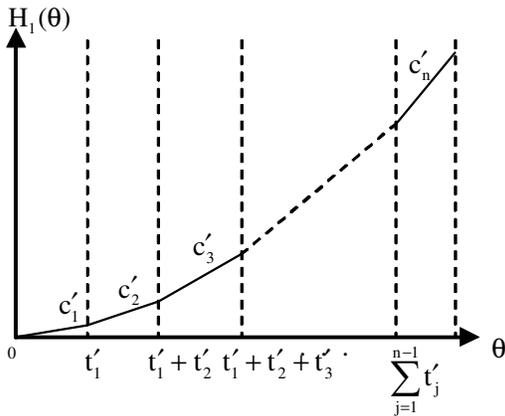


Fig. 1: Graph of  $H_1(\theta)$  of SKPCRC

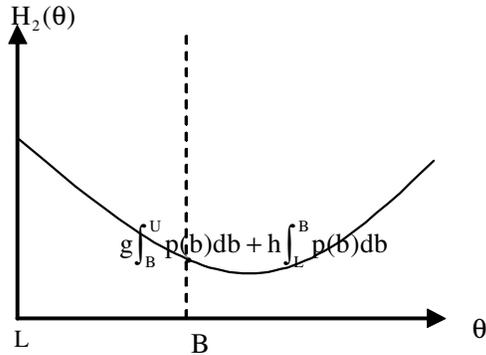


Fig. 2: Graph of  $H_2(\theta)$  of SKPCRCR

Let  $\sum_{j=1}^n y_j = \theta_{jj}, jj=1,2,\dots,n$  and  $SH1_{(t'_{j-1} \leq \theta \leq t'_j)} = \text{slope of graph } H_1(\theta), t'_{j-1} \leq \theta \leq t'_j, = c'_j$   
 Since  $c'_j \leq c'_{j+1}$ , for  $j = 1, \dots, n-1$ ,  $H_1(\theta)$  is a piecewise-linear convex function.  
 Let  $SH2_{(\theta=B)} = \text{slope of graph } H_2(\theta), \theta=B$ .  
 For  $L \leq \theta \leq U, v(b) = \theta - b, L \leq b < \theta$   
 $u(b) = b - \theta, \theta < b \leq U \quad v(b) = 0, \theta \leq b \leq U$  and  $u(b) = 0, L \leq b \leq \theta$ .  
 $H_2(\theta) = \int_{\theta}^U gp(b)(b - \theta)db + \int_L^{\theta} hp(b)(\theta - b)db$   
 $SH2_{(L \leq \theta \leq U)} = -g \int_{\theta}^U p(b)db + h \int_L^{\theta} p(b)db = -g + (g+h) \int_L^{\theta} p(b)db$   
 Since  $SH2_{(\theta)} \leq SH2_{(\theta+\Delta B)}$ , for  $L \leq \theta \leq U - \Delta B$ ,  $H_2(\theta)$  is a convex function.  
 Since  $H_1(\theta)$  is a piecewise-linear convex function and  $H_2(\theta)$  is a convex function,  $H_1(\theta) + H_2(\theta)$  is also a convex function.

Let  $SH_{\theta} = \text{slope of graph } H_1(\theta) + \text{slope of graph } H_2(\theta)$   
 $= c'_j - g \int_{\theta}^U p(b)db + h \int_L^{\theta} p(b)db$   
 The optimal solution can be found by searching for the point that has  $SH_{\theta} = 0$ . This is the same criteria as in algorithm 1 of SKPCRCR.

**Monte carlo simulation:** Monte Carlo simulation is a natural alternative for using in stochastic programs and can be applied to solve SKPCRCR. In this approach, a numerous number of samples is generated. Then each sample is solved directly as a deterministic problem. The optimal solution is the expected value of all solutions.

For each sample, the problem is as follows.

$$\text{Minimize } f = \sum_{j=1}^n c_j x_j + gu + hv$$

$$\text{Subject to } \sum_{j=1}^n a_j x_j + u - v = b \quad 0 \leq x_j \leq t_j$$

For the above model,  $v$  is always zero because the objective value of solution that have  $v > 0$  can be smaller by reducing the value of  $x_j$ . Therefore, the problem can be reduced to the following problem.

$$\text{Minimize } f = \sum_{j=1}^n c_j x_j + gu$$

$$\text{Subject to } \sum_{j=1}^n a_j x_j + u = b \quad 0 \leq x_j \leq t_j$$

**Heuristics for solving SKPDRC and SKPCRC:** The reasons for using heuristics for SKPDRC and SKPCRC are to save computing time and to provide the optimal solution or near-optimal solution. In this work, we adapted the approach used successfully in the discrete case as presented in [14,15] resulting as follows. If all elements in  $x_{SKPDRC}^*$  are integer, then the solution of SKPDRC is the same as the solution of SKPDRCR (i.e.,  $x_{SKPDRC}^* = x_{SKPDRCR}^*, u_{SKPDRC}^* = u_{SKPDRCR}^*, v_{SKPDRC}^* = v_{SKPDRCR}^*$  and  $f_{SKPDRC}^* = f_{SKPDRCR}^*$ ). Similarly, if all elements in  $x_{SKPCRC}^*$  are integer, then the solution of SKPCRC is the same as the solution of SKPCRCR (i.e.,  $x_{SKPCRC}^* = x_{SKPCRCR}^*, u_{SKPCRC}^* = u_{SKPCRCR}^*, v_{SKPCRC}^* = v_{SKPCRCR}^*$  and  $f_{SKPCRC}^* = f_{SKPCRCR}^*$ ). Otherwise, the solution of SKPDRC or SKPCRC can be found by using the proposed heuristics, which use the optimal capacity of the relaxed problem, i.e.,  $b_{SKPDRC}^*$  for discrete capacity and  $B_{SKPCRCR}^*$  for continuous capacity, as the upper bound of capacity. Due to SKPDRCR is a piecewise

convex function and SKPCRCR is a convex function, a good or near optimal solution should have the weight sum  $(\sum_{j=1}^n a_j x_j)$  near optimal capacity of the relaxed problem. The heuristics start from rounding down  $x_{SKPDRCR}^*$  or  $x_{SKPCRCR}^*$  and the upper bound of the decision variable (t). Let x is the rounding down of  $x_{SKPDRCR}^*$  or  $x_{SKPCRCR}^*$ , tint is the rounding down of t, k is the  $k_{SKPDRCR}^*$  or  $k_{SKPCRCR}^*$  and  $f_{min}$  is equal to infinity. Then calculate the weight sum and the objective value and keep the minimum objective value ( $f_{min}$ ), the solution of the SKPDRC ( $x_{int}$ ,  $u_{int}$ ,  $v_{int}$ ) or the solution of the SKPCRC ( $x_{int}$ ). Next, go to the capacity checking point. If the weight sum is less than the optimal capacity of the relaxed problem, then go to the upper bounded checking point. Otherwise, integer solution is found. For the upper bound checking point, if  $x_k$  is less than its upper bound ( $t_{int_k}$ ), then let  $x_k = x_k + 1$  and calculate the weight sum, the slack variable (u), the surplus variable (v) and the objective value (f). Otherwise, let  $k = k+1$  and  $x_k = 1$  and then calculate the weight sum, the slack variable (u), the surplus variable (v) and the objective value (f).

Next, go to the objective value checking point. If the objective value is less than  $f_{min}$ , then the new solution is found. Therefore, the old solution is replaced by the new solution. And then go back to the capacity checking point. Otherwise, keep the old solution and go back to the capacity checking point. And continue these steps until the weight sum is greater than or equal to the optimal capacity of the relaxed problem. Finally, the solution of SKPDRC or SKPCRC is found.

**RESULTS**

The experiments for SKPCRCR and SKPCRC were conducted by varying m and n. The computing time (excluding parameter generating) and solutions obtained were collected and compared. The input data were generated as follows.  $c_j, a_j, t_j$ , for  $j=1,2,\dots,n$ , were generated with uniform[0,10]. bw was normally distributed where mean was generated with uniform[0,  $\sum_{j=1}^n a_j t_j$ ] and standard deviation was generated with uniform[0, n]. G, h were generated with uniform[0,10].

$$\Delta B = 0.01 \sum_{j=1}^n a_j t_j, 0.001 \sum_{j=1}^n a_j t_j, 0.0001 \sum_{j=1}^n a_j t_j, 0.00001 \sum_{j=1}^n a_j t_j$$

The reasons for generating  $c_j, a_j, t_j, g$ , and h via uniformly distribution [0,10] were to get the various positions of the optimal solution and the optimal objective function value will not be too large. Moreover, the reason for generating b with normally distributed where mean was generated with uniform  $[0, \sum_{j=1}^n a_j t_j]$  and standard deviation was generated with

uniform [0, n] was to ensure that all items fit in the capacity. For stepping size ( $\Delta B$ ), we conducted the experiments with four values of  $\Delta B$  in order to select the appropriate  $\Delta B$  for this studied problem. According to the obtained results, the computing time (sec) of the algorithm 1 (or algorithm 2) was smaller than the computing time of the Monte Carlo simulation, when

$$\Delta B = 0.01 \sum_{j=1}^n a_j t_j \text{ and } 0.001 \sum_{j=1}^n a_j t_j \text{ for all } n \text{ and when}$$

$$\Delta B = 0.0001 \sum_{j=1}^n a_j t_j, \text{ the computing time (sec) of the}$$

algorithm 1 (or algorithm 2) was smaller than the computing time of the Monte Carlo simulation for  $n \leq 2500$ . Furthermore, for  $\Delta B = 0.01 \sum_{j=1}^n a_j t_j$  and

$$0.001 \sum_{j=1}^n a_j t_j, \text{ the percent winning in computing time is}$$

100% for all n. And the percent winning decreased when  $\Delta B$  is more delicate and also decrease when n increased. Again, fifty generated samples were conducted to compare the quality of solutions between using the algorithm 1 (or algorithm 2) and the Monte Carlo simulation for solving the relaxed problem. From

the results, when  $\Delta B$  was  $0.01 \sum_{j=1}^n a_j t_j$ , the percent of win

was 76%. And the percent of win was improved to more than 95% when  $\Delta B$  was more delicate. In

addition, when  $\Delta B$  was  $0.00001 \sum_{j=1}^n a_j t_j$ , the percent of

win was 100%. The percent of win were more than 50% for all  $\Delta B$ . And when  $\Delta B$  was  $0.001 \sum_{j=1}^n a_j t_j$ ,

$$0.0001 \sum_{j=1}^n a_j t_j \text{ and } 0.00001 \sum_{j=1}^n a_j t_j, \text{ the percent of not}$$

losing (either win or tie) were more than 95%. Furthermore, when n increased the percent of win tended to be increased. The average absolute percent error was calculated in order to represent how far of the

objective value of the algorithm 1 (or algorithm 2) compared with the Monte Carlo simulation. According

to the obtained result, for  $\Delta B = 0.01 \sum_{j=1}^n a_j t_j$ , the values of average absolute percent error of the winning cases and losing cases were not much different. However, for  $\Delta B = 0.001 \sum_{j=1}^n a_j t_j$  and  $0.0001 \sum_{j=1}^n a_j t_j$ , the average absolute percent error of the winning cases were significantly greater than the average absolute percent error of the losing cases, i.e. for  $\Delta B = 0.00001 \sum_{j=1}^n a_j t_j$ , there was no losing case at all. Moreover, for  $\Delta B = 0.001 \sum_{j=1}^n a_j t_j$ ,  $0.0001 \sum_{j=1}^n a_j t_j$  and  $0.00001 \sum_{j=1}^n a_j t_j$ , the values of the average absolute percent error of the winning cases tended to increase when  $n$  increased. If scale of  $\Delta B$  was more delicate, the average absolute percent error of losing cases significantly decreased but the average absolute percent error of the winning cases only slightly decreased. When  $\Delta B = 0.01 \sum_{j=1}^n a_j t_j$ , the values of the average absolute percent error of the winning cases were significantly greater than the average absolute percent error of the losing cases for small  $n$  (i.e.,  $n \leq 500$ ). However, the values of the average absolute percent error of the winning cases were only slightly greater than the average absolute percent error of the losing case for large  $n$  (i.e.,  $n \geq 750$ ). When  $\Delta B = 0.001 \sum_{j=1}^n a_j t_j$ ,  $0.0001 \sum_{j=1}^n a_j t_j$  and  $0.00001 \sum_{j=1}^n a_j t_j$ , the values of the average absolute percent error of the winning case were significantly greater than the average absolute percent error of the losing cases for all  $n$ . If scale of  $\Delta B$  was more delicate, the average absolute percent error of the losing cases tended to be decreased.

### DISCUSSION

In order to select the appropriate algorithm, the computing time and the quality of the solutions are important criteria. From all results, it can be concluded that the algorithm 1 (or algorithm 2) with appropriate  $\Delta B$  was superior to the Monte Carlo simulation. The ratios of computing time was calculated from taking the computing time of the algorithm 1 (or algorithm 2) of SKPCRCR divided by the computing time of the Monte Carlo simulation for solving SKPCRCR. The ratio of computing time was calculated from taking the computing time of the proposed heuristic where the

relaxed problem was solved by using the algorithm 1 (or algorithm 2) of SKPCRCR divided by the computing time of the proposed heuristic where the relaxed problem was solved by using the Monte Carlo simulation. For  $n = 5,000$ , if the quality of the solutions is more important criteria than the computing time, then better alternative is  $\Delta B = 0.0001 \sum_{j=1}^n a_j t_j$ . The computing time of this  $\Delta B$  was about 57.5 seconds more than the computing time of the Monte Carlo simulation for both SKPCRCR and SKPCRC. However, the percent of win of SKPCRCR was increased from 98-100% while the percent of lose of SKPCRCR was decreased from 2-0%. In addition, the percent of win of SKPCRC was increased from 82-84% while the percent of lose of SKPCRC was decreased from 10-4%.

### CONCLUSION

The approaches for solving stochastic knapsack problem with continuous random capacity were presented. For the proposed procedure, the relaxed integer constraints were employed. Next we solved the relaxed problem. Then the optimal solution of the relaxed problem was found as the initial solution for finding the final integer solution. Two algorithms were proposed for solving the relaxed problem (SKPCRCR) and compared with the Monte Carlo simulation. According to the proof of the algorithms 1 and 2 of SKPCRCR, the criteria of these two algorithms were the same. The fifty generated samples were generated to compare an efficiency (i.e., computing time) and effectiveness (i.e., quality of solutions) between the algorithm 1 (or algorithm 2) and the Monte Carlo simulation. For the algorithm 1 (or algorithm 2) of SKPCRCR, if  $\Delta B$  was more delicate, the computing time increased but it gave the better solution. As the results, the algorithm 1 (or algorithm 2) of SKPCRCR with appropriate  $\Delta B$  was superior to the Monte Carlo simulation of SKPCRCR

For SKPCRC where the relaxed problem was solved by using the algorithm 1 (or algorithm 2) of SKPCRCR, if  $\Delta B$  was more delicate, the computing time increased but it gave better solution. As the results, the proposed heuristic for solving SKPCRC where the relaxed problem was solved by using the algorithm 1 (or algorithm 2) of SKPCRCR with appropriate  $\Delta B$  was superior to the proposed heuristic for solving SKPCRC where the relaxed problem was solved by using the Monte Carlo simulation

In addition, the study for characteristic types of generated data, which make the integer solution of the

proposed heuristic for SKPCRC where the relaxed solution was solved by using the algorithm 1 (or algorithm 2) of SKPCRCR is worse than integer solution of the proposed heuristic for SKPCRC solved by using the Monte Carlo simulation also should be investigated. Furthermore, it is worth to study the other types of stochastic knapsack problem such as both uncertain benefit and capacity and both weight and capacity.

## REFERENCES

1. Martello, S. and P. Toth, 1990. Knapsack problems: Algorithm and computer implementations. 1st Edn., John Wiley and Sons, West Sussex, England, pp: 296. ISBN:0-471-92420-2.
2. Parks, M.S. and E. Steinberg, 1979. A preference order dynamic program for a knapsack problem with stochastic rewards. *J. Operat. Res. Soc.*, 30: 141-147. DOI: 10.1057/jors.1979.27.
3. Sniedovich, M., 1980. Preference order stochastic knapsack problems: Methodological issues. *J. Operat. Res. Soc.*, 31: 1025-1032. DOI:10.1057/jors.1980.189.
4. Henig, M.I., 1990. Risk criteria in a stochastic knapsack problem. *Operat. Res.*, 38: 820-825. <http://portal.acm.org/citation.cfm?id=88379.88401>.
5. Kleywegt, A.J. and J.D. Papastavrou, 1998. The dynamic and stochastic knapsack problem. *Operat. Res.*, 46: 17-35. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.7.5029>.
6. Chiu, L.A., Jr. Cox and L.L. Lu, 1999. Optimal project selection: Stochastic knapsack with finite time horizon. *J. Operat. Res. Soc.*, 50: 645-650. DOI: 10.1057/palgrave.jors.2600721.
7. Kleywegt, A.J., J.D. Papastavrou and S. Rajagopalan, 1996. The dynamic and stochastic knapsack problem with deadlines. *Manage. Sci.*, 42: 1706-1718. DOI: 10.1287/mnsc.42.12.1706.
8. Slyke, R.V. and Y. Young, 2000. Finite horizon stochastic knapsacks with applications to yield management. *Operat. Res.*, 48: 155-172. <http://portal.acm.org/citation.cfm?id=767739.768671>.
9. Kleywegt, A.J. and J.D. Papastavrou, 2001. The dynamic and stochastic knapsack problem with random sized items. *Operat. Res.*, 49: 26-41. DOI: 10.1287/opre.49.1.26.11185.
10. ILOG, 2008. CPLEX 8.1. <http://www.ilog.com/products/cplex>.
11. Witchakul, S. and P. Sudasna-na-Ayudthya, 2005. Stochastic knapsack problem with continuous random capacity. *J. Res. Eng. Technol.*, 2: 339-376.
12. Witchakul, S., P. Sudasna-na-Ayudthya and P. Charnsethikul, 2007. A fast method for a class of one-stage bounded variables and single constrained linear programming problems. *J. Interdisciplinary Math.*, 10: 285-303. [http://www.connectjournals.com/toc.php?aid=P.%20CHARNSETHIKUL&&bookmark=CJ-003060&&volume=10&&issue\\_id=2&&issue\\_month=April&&year=2007](http://www.connectjournals.com/toc.php?aid=P.%20CHARNSETHIKUL&&bookmark=CJ-003060&&volume=10&&issue_id=2&&issue_month=April&&year=2007).
13. Witchakul, S., P. Sudasna-na-Ayudthya, P. Charnsethikul and K.A. Mathur, 2007. A graphical based method for a class of one-stage bounded variables and single constrained linear programming problems. *Am. J. Appl. Sci.*, 4: 1040-1044. <http://www.scipub.org/fulltext/ajas/ajas4121040-1044.pdf>.
14. Witchakul, S., P. Sudasna-na-Ayudthya and P. Charnsethikul, 2007. A heuristic for solving a stochastic knapsack problem with discrete random capacity. *Thammasat Int. J. Sci. Technol.*, 12: 1-9. [http://www.tijsat.tu.ac.th/issues/2007/no1/2007\\_V12\\_No1\\_1.PDF](http://www.tijsat.tu.ac.th/issues/2007/no1/2007_V12_No1_1.PDF).
15. Witchakul, S., P. Sudasna-na-Ayudthya and P. Charnsethikul, 2007. Solving stochastic knapsack problem with discrete random capacity. *Proceedings of 37th Conference in Computer and Industrial Engineering*, Oct. 20-23, alexandria, Egypt, pp: 1199-1208. [http://www0.umoncton.ca/cie/conferences/37thconf/CIE37%20Proceedings/Directory\\_Files/Files/Papers/cie00168cie00168ICCIE37\\_PaperTemplate\\_Suwitchaporn\\_Witchakul\\_revised.pdf](http://www0.umoncton.ca/cie/conferences/37thconf/CIE37%20Proceedings/Directory_Files/Files/Papers/cie00168cie00168ICCIE37_PaperTemplate_Suwitchaporn_Witchakul_revised.pdf).