

A Bayesian Optimisation Approach for Multidimensional Knapsack Problem

Hanyu Gu^[0000–0003–2035–2583], Alireza Etmianiesfahani, and Amir Salehipour

School of Mathematical and Physical Sciences, University of Technology Sydney
15 Broadway Ultimo NSW 2007, Australia

Hanyu.Gu@uts.edu.au, Alireza.Etmianiesfahani@student.uts.edu.au,
amir.salehipour@uts.edu.au

Abstract. This paper considers the application of Bayesian optimisation to the well-known multidimensional knapsack problem which is strongly NP-hard. For the multidimensional knapsack problem with a large number of items and knapsack constraints, a two-level formulation is presented to take advantage of the global optimisation capability of the Bayesian optimisation approach, and the efficiency of integer programming solvers on small problems. The first level makes the decisions about the optimal allocation of knapsack capacities to different item groups, while the second level solves a multidimensional knapsack problem of reduced size for each item group. To accelerate the Bayesian optimisation guided search process, various techniques are proposed including variable domain tightening, initialisation by the Genetic Algorithm, and optimisation landscape smoothing by local search. Computational experiments are carried out on the widely used benchmark instances with up to 100 items and 30 knapsack constraints. The preliminary results demonstrate the effectiveness of the proposed solution approach.

Keywords: Bayesian optimisation · Multidimensional knapsack problem · Meta-heuristics.

1 Introduction

The Bayesian optimisation (BO) is a powerful machine learning based method for the optimisation of expensive black-box functions, which typically only allow point-wise function evaluation [23, 22]. Although BO has been widely used in the experimental design community since the 1990s [15, 13], it is not until the last decade that BO has become extremely popular in the machine learning community as an efficient tool for tuning hyper-parameters in various algorithms, e.g., deep learning [5, 7], natural language processing [29], and preference learning [10]. The BO is also embraced by new areas such as robotics [16], automatic control [1], and pharmaceutical product development [21].

The Multidimensional Knapsack Problem (MKP) is an extension of the classic Knapsack Problem (KP). It comprises of n items and m knapsacks with limited capacities. Each item contributes a certain amount of profit if selected

and consumes “resources” simultaneously in each knapsack. The MKP aims for a subset of items that achieves the highest total profit while abiding by the capacities of all knapsacks. The MKP is a well-known, and strongly NP-hard combinatorial optimisation problem, and has found applications in many practical areas involving resource allocation [11, 17]. In spite of the tremendous progress made in exact solution techniques, many instances from the widely used Chu and Beasley MKP test set [4] cannot be solved to optimality [8, 12, 28], especially when the number of knapsacks is large. The best known solutions on the hard instances are all obtained by specialised meta-heuristics which require exorbitant computation time [24, 25, 27, 3, 6]. The simplicity of problem statement and computational hardness makes the MKP an ideal test bed for new solution ideas and techniques [14, 18].

The BO encounters insurmountable issues to solve the MKP. Firstly, the BO is designed to solve problems with simple feasible set of continuous variables [9], while the MKP has only binary variables with many knapsack constraints. Whereas a lot of efforts have been committed to consider feasible set with combinatorial structures, all the reported computational studies investigated problems with just a few dozen categorical/integer/binary variables [2, 19]. Secondly, the BO is only efficient for low dimensional problems with less than 20 variables, while the MKP can have hundreds of binary variables. Although the BO with random embedding can solve problems with billions of variables, it relies on the “low effective dimensionality” which can be an issue for MKP [26]. Finally, the MKP has a linear function which is “cheap” to calculate, which makes it hard for the BO to compete with other meta-heuristic and artificial intelligence algorithms.

Based on the idea of divide and conquer, a novel two-level model for MKP (TL-MKP) is proposed in this paper to take advantage of the special structure of MKP, i.e., the number of items (variables) is much larger than the number of knapsacks (constraints). In particular, the items are divided into groups, and the knapsack capacities allocated to each group are determined by the first level, or master problem, of the TL-MKP. With assigned knapsack capacities, each group can be solved as a MKP of reduced size in the second-level of TL-MKP, or subproblem. It is shown in Section 2 that the master problem has a non-continuous, multi-modal, and expensive to evaluate objective function with simple feasible set, which is suitable for the application of BO. Since the subproblem has a much smaller number of binary variables, it can be efficiently solved to optimality with commercial integer programming solvers.

It is essential to incorporate prior knowledge in the BO, which was designed to be a black-box global optimisation method. Two novel techniques are presented in this paper to make use of the information provided by mathematical programming solver and meta-heuristics. Indeed, when a good solution is known, e.g., by using other meta-heuristics, an efficient heuristic is proposed in this paper to tighten the domain bounds of the master problem in the TL-MKP. Inspired by the simulation approach used in robotics control algorithms to initialise the BO [20], the Genetic Algorithm (GA) is used in this paper to generate initial

trial points for the BO. To take advantage of the linear structure of the objective function of MKP, the GA is run on the MKP instead of the master problem of TL-MKP. These techniques can significantly accelerate the search process of BO as demonstrated by the computational experiments in this paper.

The paper is organised as follows. The novel two-level model for MKP is presented with discussion of the properties of the master problem in Section 2. The BO based optimisation approach and some acceleration techniques are described in Section 3. The implementation details are discussed in Section 4. Computational results are presented in Section 5. The conclusion is given in Section 6.

2 Two-level Model for MKP

Given m knapsacks with capacities b_i , $i = 1, \dots, m$, and a set of n items $I = \{1, 2, \dots, n\}$, each item j requires a resource consumption of $a_{i,j}$ units in the i -th knapsack, $i = 1, \dots, m$, and yields c_j units of profit upon inclusion, $j = 1, \dots, n$. The goal is to find a subset of items that yields maximum profit, denoted by z^* , without exceeding the knapsack capacities. The MKP can be defined by the following integer linear programming model:

$$(\text{MKP}) \quad z^* = \max\{c^T x : Ax \leq b, x \in \{0, 1\}^n\}, \quad (1)$$

where $c = [c_1, c_2, \dots, c_n]^T$ is an n -dimensional vector of profits, $x = [x_1, x_2, \dots, x_n]^T$ is an n -dimensional vector of 0-1 decision variables indicating whether an item is included or not, $A = [a_{i,j}]$, $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$ is an $m \times n$ coefficient matrix of resource requirements, and $b = [b_1, b_2, \dots, b_m]^T$ is an m -dimensional vector of resource capacities. It is further assumed that all parameters are non-negative integers.

Assume the items are divided into two groups, i.e., $I = I_1 \cup I_2$, and $I_1 \cap I_2 = \emptyset$. Each group is formulated as a MKP with profit vector $c^i = c_{I_i}$, resource requirement matrix $A^i = A_{I_i}$, and capacity vector $b^i \in \mathbb{R}^m$. The two groups share the capacities of the m knapsacks, i.e.,

$$b^1 + b^2 = b \quad (2)$$

The first level of the TL-MKP (the two-level model for MKP), or the master problem is defined as

$$(\text{L1-MKP}) \quad f^* = \max\{f(t) : t \in \mathbb{R}^m, 0 \leq t \leq b\}, \quad (3)$$

where

$$f(t) = z_1^*(t) + z_2^*(b - t) \quad (4)$$

is calculated by solving the second level of the TL-MKP, or subproblems:

$$(\text{L2-MKP}) \quad z_i^*(u) = \max\{c^i x^i : A^i x^i \leq u, x^i \in \{0, 1\}^{|I_i|}\}, \quad i = 1, 2 \quad (5)$$

Since each solution of the TL-MKP can be easily converted to a solution to the MKP with the same objective value, and each solution of the MKP can be used to define a value of t for the master problem of TL-MKP (3), the following proposition holds.

Proposition 1. *($t^* = A^1 x^{1*}, x^{1*}, x^{2*}$) is an optimal solution of TL-MKP if and only if x^* , defined as $x_{N_1}^* = x^{1*}$ and $x_{N_2}^* = x^{2*}$, is an optimal solution of MKP. Furthermore, $f^* = z^*$.*

Example 1. Consider an instance of MKP with three items and one knapsack, where $c = [1, 2, 3]$, $A = [1, 2, 3]$, and $b = 4$. The two groups are $I_1 = \{1, 2\}$ and $I_2 = \{3\}$. It is straightforward to show that the first level objective function is

$$f(t) = \begin{cases} 3 & t \in [0, 1) \\ 4 & t = 1 \\ 1 & t \in (1, 2) \\ 2 & t \in [2, 3) \\ 3 & t \in [3, 4] \end{cases}.$$

Example 2. Consider an instance of MKP with 20 items and two knapsacks. The two groups have the same number of items. Fig. 1 shows the contour graph of the first level objective function $f(t)$. The optimal value is equal to 75.

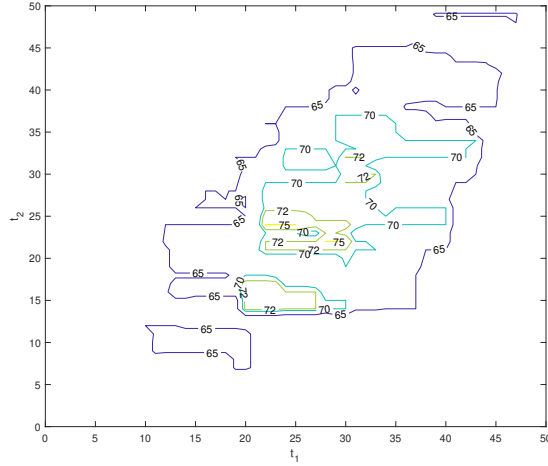


Fig. 1. Contour of the first level objective function $f(t)$; $t_1(t_2)$ is the capacity allocated to group 1 (2).

Examples (1) and (2) clearly demonstrate that the objective function of the master problem in TL-MKP is non-continuous, and can have many local optima. Although the subproblems have much smaller sizes, they are still more expensive to evaluate than the linear function of MKP.

It can be observed that $f(t)$ is not differentiable when at least one knapsack has no slack capacity in one of the subproblems. That leads to the following proposition,

Proposition 2. *$f(t)$ is differentiable almost everywhere in the sense of Lebesgue measure with $f'(t) = 0$.*

Although $f(t)$ is differentiable almost everywhere, the derivative is constantly zero and consequently, useless for the design of optimisation algorithms.

3 Bayesian Optimisation and Acceleration

The BO is a promising option to deal with the challenges presented by the master problem of TL-MKP such as no closed form, non-continuity, multiple local optima, absence of useful derivatives, and high cost of function evaluation. In this section, the basic principles of BO are described first [9], then followed by techniques to incorporate prior knowledge to accelerate the search process.

The BO builds a probabilistic model for the unknown $f(t)$ of the master problem of TL-MKP. In particular, $f(t)$ is assumed to be drawn from a Gaussian process (GP), which is determined by a mean function $\mu_0 : \mathbb{R}^m \rightarrow \mathbb{R}$, and a positive definite covariance function $k_0 : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}$, also known as the kernel of the GP. The BO sequentially generates points to evaluate within the feasible region of TL-MKP. Assume that n points have been evaluated with observations $\mathcal{D}_n = \{(t^1, f(t^1)), (t^2, f(t^2)), \dots, (t^n, f(t^n))\}$. Using Bayes' rule, the conditional distribution of $f(t)$ is derived as a Normal distribution:

$$P(f(t)|\mathcal{D}_n, t) = \mathcal{N}(\mu_n(t), \sigma_n^2(t)) \quad (6)$$

$$\mu_n(t) = \Sigma_0(t, t_{1:n})\Sigma_0(t_{1:n}, t_{1:n})^{-1}(f(t_{1:n}) - \mu_0(t_{1:n})) + \mu_0(t) \quad (7)$$

$$\sigma_n^2(t) = k_0(t, t) - \Sigma_0(t, t_{1:n})\Sigma_0(t_{1:n}, t_{1:n})^{-1}\Sigma_0(t_{1:n}, t) \quad (8)$$

where $f(t_{1:n}) = [f(t^1), \dots, f(t^n)]^T$, $\mu_0(t_{1:n}) = [\mu_0(t^1), \dots, \mu_0(t^n)]^T$, and

$$\Sigma_0(t_{1:n}, t_{1:n}) = \begin{pmatrix} k_0(t^1, t^1) & \dots & k_0(t^1, t^n) \\ \vdots & \ddots & \vdots \\ k_0(t^n, t^1) & \dots & k_0(t^n, t^n) \end{pmatrix}.$$

The BO selects the next most promising point to evaluate, i.e., t^{n+1} , by optimising an acquisition function, which balances exploration (uncertainty $\sigma_n(t^{n+1})$ is large) against exploitation (objective expected value $\mu_n(t^{n+1})$ is large). Different types of acquisition function have been proposed in the literature, while the most commonly used is *Expected Improvement* (EI). The EI acquisition function is defined as

$$\mathbf{EI}_n(t) = E_n(\max(f(t) - \max_{i=1}^n f(t^i), 0)), \quad (9)$$

where $E_n(\cdot)$ is the expectation taken under the posterior distribution (6).

The next point to evaluate is selected as

$$t^{n+1} = \operatorname{argmax}_t \mathbf{EI}_n(t). \quad (10)$$

With new point $(t^{n+1}, f(t^{n+1}))$, the conditional probability of $f(t)$ can be updated according to (6), and the iterative process stops when a sampling budget is reached.

3.1 Variable Domain Tightening

The efficiency of BO depends on the size and dimensionality of the search space of TL-MKP, which is defined in (3) as $[0, b] \subset \mathbb{R}^m$. If a good lower bound of MKP f_L is known, e.g., through a quick meta-heuristic, the search space can be reduced to $F = \{t | f(t) \geq f_L, t \in [0, b] \subset \mathbb{R}^m\}$. However, this will make the EI acquisition function harder to optimise in (10) since F has no simple representation. In this paper, an optimisation based approach is employed to find the smallest hypercube $H = [t^L, t^U]$ that contains F , i.e., $F \subset H$. The upper bound of H along the i -th coordinate, t_i^U , $i = 1, \dots, m$, can be obtained by solving

$$t_i^U = \max\{A^1 x^1 : c^T x \geq f_L, Ax \leq b, x^1 = x_{I_1}, x \in \{0, 1\}^n\}. \quad (11)$$

The lower bound of H along the i -th coordinate, t_i^L , $i = 1, \dots, m$, can be obtained by solving

$$t_i^L = \min\{A^1 x^1 : c^T x \geq f_L, Ax \leq b, x^1 = x_{I_1}, x \in \{0, 1\}^n\}. \quad (12)$$

The exact solution of (11) and (12) is time-consuming. Therefore, t^U (t^L) can be replaced by a upper (lower) bound of (11) ((12)), e.g., using the linear programming relaxation by replacing $x \in \{0, 1\}^n$ with $x \in [0, 1]^n$.

3.2 Initialisation with Genetic Algorithm

The BO randomly generates the initial trial points in the search space which can lead to slow convergence. In this paper, The GA is used to generate initial points that have good solution quality as well as diversity in the search space. The GA is a population based meta-heuristic which evolves by generations through genetic operators such as cross-over and mutation. In the early stage of GA the population has good diversity but low percentage of good solutions; while in the later stage, the population has high percentage of good solutions but with less diversity.

It is computationally infeasible to run GA on the TL-MKP since the objective evaluation involves solving two MIP problems and consequently expensive. Instead, the GA is directly run on the MKP, and the population is mapped to initialise the BO for TL-MKP. In particular, let \tilde{x} be a solution from a population of GA. The mapped solution for TL-MKP becomes

$$\tilde{t} = A^1 \tilde{x}_{N_1}. \quad (13)$$

It is easy to see that

$$f(\tilde{t}) \geq c^T \tilde{x}. \quad (14)$$

3.3 Optimisation Landscape Smoothing

At each sampling point of BO, a feasible solution to the MKP is also generated according to Proposition 1. This solution can be improved by a local search which is efficient to cope with large number of items and constraints. We define the neighbourhood of a solution x as the set of solutions with at most k different items:

$$N_k(x) = \{y \in \{0, 1\}^n : Ay \leq b, \|x - y\|_1 \leq k\}. \quad (15)$$

For Example 1, with $k = 1$, the first level objective function becomes

$$f(t) = \begin{cases} 4 & t \in [0, 2) \\ 3 & t \in [2, 4] \end{cases},$$

which is "smoother" in terms of the optimisation landscape.

4 Implementation

The BO approach for the MKP (BO-MKP) can be described as in Alg. 1, and a prototype of BO-MKP was implemented in Matlab R2020b. In Step 1 of BO-MKP, the linear relaxation of (11) and (12) are solved to tighten the bounds of the feasible set of TL-MKP using the function *linprog* in Matlab *Optimization Toolbox*. Using the function *ga* in the *Global Optimization Toolbox* of Matlab, an initial set of trial points are generated in Step 2 as input for BO according to (13). In Step 3, the BO is implemented with the function *bayesopt* in the *Global Optimization Toolbox* of Matlab. The acquisition function is set to "expected-improvement", and the maximum number of evaluation, "MaxObjectiveEvaluations", is set to N which is a user specified parameter. The subproblems of TL-MKP (5) are solved by the mixed integer programming solver *intlinprog* in Matlab *Optimization Toolbox*. In Step 4, The best solution of TL-MKP found by BO is converted to a solution of MKP with the same objective function value according to Proposition 1.

The selection of kernel function for GP can have a strong influence on the performance of BO. *bayesopt* uses the ARD Matérn 5/2 kernel

$$k(x_i, x_j | \sigma_f, \sigma_l) = \sigma_f^2 \left(1 + \frac{\sqrt{5}r}{\sigma_l} + \frac{5r^2}{3\sigma_l^2} \right) \exp \left(-\frac{\sqrt{5}r}{\sigma_l} \right)$$

where $r = \sqrt{(x_i - x_j)^T (x_i - x_j)}$, and the parameters are estimated by Gaussian process regression *fitrgp*.

Algorithm 1: The BO approach for the MKP (BO-MKP).

Input: MKP, item groups I_1 and I_2 , lower bound of MKP f_L ,
maximum number of evaluation N for BO.
Output: a feasible solution of MKP.
Step 1: tighten the bounds of feasible set of TL-MKP based on f_L ;
Step 2: generate initial trial points using GA;
Step 3: search for the global optimum of TL-MKP using BO within a
sampling budget of N evaluations;
Step 4: convert the best solution found by BO to the solution of MKP;
return

5 Computational Experiments

All experiments are carried out on the widely used Chu and Beasley MKP test set in [4]. The Chu and Beasley test set contains classes of randomly generated instances for each combination of $n \in \{100, 250, 500\}$ items, $m \in \{5, 10, 30\}$ constraints, and tightness ratios $\alpha \in \{0.25, 0.5, 0.75\}$ with smaller α representing tighter resource capacities. In the Chu and Beasley MKP test set, the resource consumption values a_{ij} are integers uniformly chosen from $(0, 1000)$, which leads to large values of the knapsack capacities b . Since the search space of BO for TL-MKP is defined by b in (3), the Chu and Beasley MKP test set is an challenging test bed for the proposed BO approach.

To show the effect of tightening bounds in section 3.1, the BO is tested on three selected instances with $n = 100$, and the results are reported in Table 1. The rows correspond to the instances with the number of knapsack constraints $m = 5, 10$ and 30 . The optimal values of these instances are obtained by CPLEX and reported in the column titled “Opt.” The columns are divided into two groups for the BO results, one for the cases without bound tightening (“Without tightening”) and the other one for the cases with bound tightening (“With tightening”). To have a better understanding of the convergence behavior of BO, two values are applied for the maximum number of evaluations, i.e., $N = 25, 50$. Since the BO is a stochastic algorithm, the average objective function value of 5 runs is reported for each pair of (m, N) in the columns titled “Ave.”. The relative gap for the solution found by the BO is calculated as $100 \times (z^* - f)/z^*$ and reported in the columns titled “gap(%)”. It can be seen that the performance of BO deteriorates dramatically when m increases. When $m = 30$, the BO reaches a massive relative gap of 63.8% after 50 function evaluations. This observation is consistent with BO’s behavior for other optimisation problems. When bound tightening technique is applied, the performance of BO is improved on all (m, N) pairs. The improvement is more dramatic when m becomes large. For $m = 10$ and $N = 50$ the relative gap is reduced from 9.8% to 4%. However, the solution quality for $m = 30$ is still not satisfactory with a large gap of 31.8%.

Table 2 presents the results of BO-MKP which initialises the BO with GA. The initial trial points provided by the GA should be diverse enough while also having good solution quality. Therefore, the maximum number of iterations of

Table 1. Effects of bound tightening for BO on the TL-MKP.

		Without tightening				With tightening			
		N = 25		N = 50		N = 25		N = 50	
m	Opt.	Ave.	gap(%)	Ave.	gap(%)	Ave.	gap(%)	Ave.	gap(%)
5	24381	22897	6.1	23849	2.2	23913	1.9	24017	1.5
10	23064	17145	25.7	20806	9.8	20581	10.8	22149	4.0
30	21946	5710.2	74.0	7955	63.8	14659	33.2	14978	31.8

GA is limited to 55 in BO-MKP. It can be seen that the GA initialisation is not helpful when $m = 5$, which suggests that the BO has strong global search capability when the dimension is low. In sharp contrast, the BO-MKP dramatically reduces the relative gap for larger dimension. Indeed, the relative gap is just 4% for $m = 30$ with 50 function evaluations.

Table 2. Effects of GA initialisation for BO on the TL-MKP.

		N = 25				N = 50		
		Opt.	w/o GA	Ave.	gap(%)	w/o GA	Ave.	gap(%)
$m = 5$	24381	23913	23928	1.9	24017	24063	1.3	
$m = 10$	23064	20581	22471	2.6	22149	22396	2.9	
$m = 30$	21946	14659	20727	5.6	14978	21060	4.0	

Table 3 shows the impact of employing the local search in solving the BO-MKP. With $k = 5$ for the neighbourhood defined in (15), the three instances with $m = 5, 10$ and 30 are all solved to optimality.

Table 3. Effects of local search for BO on the TL-MKP.

		N = 25			
		Opt.	w/o LS	with LS	gap(%)
$m = 5$	24381	23928	24381	0.0	
$m = 10$	23064	22471	23064	0.0	
$m = 30$	21946	20727	21946	0.0	

The overall performance of BO-MKP on all the 90 instances with 100 items, i.e., $n = 100$ is presented in Table 4. For the groups with $m = 5$ and $m = 10$, we set $N = 25$ and $k = 5$. For all instances with $m = 5$ and 26 instances with $m = 10$, the optimal solutions are obtained. The remaining 4 instances in the group with $m = 10$ can also be solved to optimality by increasing N to 50. We set $k = 10$ and $N = 50$ for all instances with $m = 30$. This group of instances is particularly challenging to BO due to the high dimensions of

the search space. However, with a strong local search procedure to smooth the optimisation landscape, high quality solutions are obtained on all instances.

Table 4. Computational results for all instances with 100 items.

		$\alpha = 0.25$	$\alpha = 0.5$	$\alpha = 0.75$
$m = 5$	Average	24197.2	43252.9	60471.0
	Best	24197.2	43252.9	60471.0
	Opt.	24197.2	43252.9	60471.0
	gap %	0.0	0.0	0.0
	time	146.7s	128.8s	83.3s
$m = 10$	Average	22601.0	42660.2	59555.6
	Best	22601.9	42660.6	59555.6
	Opt.	22601.9	42660.6	59555.6
	gap %	0.0	0.0	0.0
	time	191.5s	195.3s	152.3s
$m = 30$	Average	21638.2	41420.3	59201.8
	Best	21652.9	41427.2	59201.8
	Opt.	21660.4	41440.4	59201.8
	gap %	0.1	0.0	0.0
	time	359.0s	359.3s	311.3s

6 Conclusion and Future Work

In this paper, a two-level model is presented for the multidimensional knapsack problem. The master problem has much smaller dimensions, which makes it amenable to Bayesian optimisation. Three techniques are introduced to accelerate the search process of BO. Preliminary test results show the effectiveness of the proposed approach. It strongly demonstrates that incorporating prior knowledge and smoothing the optimisation landscape by the local search are crucial for the success of BO for large MKP.

Future work includes the investigation of the proper kernels in BO for combinatorial optimisation problems, the automatic tuning of hyper-parameters, and comparison with other meta-heuristics. It is also interesting to extend the models to combinatorial optimisation problems with more complex structures.

References

1. Baheri, A., Bin-Karim, S., Bafandeh, A., Vermillion, C.: Real-time control using bayesian optimization: A case study in airborne wind energy systems. *Control Engineering Practice* **69**, 131 – 140 (2017).
2. Baptista, R., Poloczek, M.: Bayesian optimization of combinatorial structures. In: Dy, J., Krause, A. (eds.) *Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research*, vol. 80, pp. 462–471. PMLR, Stockholmsmässan, Stockholm Sweden (10–15 Jul 2018)

3. Boussier, S., Vasquez, M., Vimont, Y., Hanafi, S., Michelon, P.: A multi-level search strategy for the 0–1 multidimensional knapsack problem. *Discrete Applied Mathematics* **158**(2), 97–109 (2010)
4. Chu, P.C., Beasley, J.E.: A genetic algorithm for the multidimensional knapsack problem. *J. Heuristics* **4**(1), 63–86 (1998)
5. Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., aurelio Ranzato, M., Senior, A., Tucker, P., Yang, K., Le, Q.V., Ng, A.Y.: Large scale distributed deep networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems 25*, pp. 1223–1231. Curran Associates, Inc. (2012)
6. Della Croce, F., Grosso, A.: Improved core problem based heuristics for the 0/1 multi-dimensional knapsack problem. *Computers & Operations Research* **39**(1), 27–31 (2012)
7. Domhan, T., Springenberg, J.T., Hutter, F.: Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In: *Proceedings of the 24th International Conference on Artificial Intelligence*. p. 3460–3468. IJCAI’15, AAAI Press (2015)
8. Drake, J.: Or library mkn - best known solutions, <http://www.cs.nott.ac.uk/~jqd/mkp/bestresults.html>
9. Frazier, P.I.: Bayesian optimization. *INFORMS TutORials in Operations Research* p. 255–278 (2018).
10. Freno, A., Saveski, M., Jenatton, R., Archambeau, C.: One-pass ranking models for low-latency product recommendations. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. p. 1789–1798. KDD ’15, Association for Computing Machinery, New York, NY, USA (2015).
11. Fréville, A.: The multidimensional 0 - 1 knapsack problem: An overview. *European Journal of Operational Research* **155**(1), 1–21 (2004)
12. Fréville, A., Hanafi, S.: The multidimensional 0-1 knapsack problem - bounds and computational aspects. *Annals of Operations Research* **139**(1), 195–227 (2005)
13. Greenhill, S., Rana, S., Gupta, S., Vellanki, P., Venkatesh, S.: Bayesian optimization for adaptive experimental design: A review. *IEEE Access* **8**, 13937–13948 (2020)
14. Gu, H.: Improving problem reduction for 0-1 multidimensional knapsack problems with valid inequalities. *Comput. Oper. Res.* **71**(C), 82–89 (Jul 2016).
15. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. *Journal of Global optimization* **13**(4), 455–492 (1998)
16. Junge, K., Hughes, J., Thuruthel, T.G., Iida, F.: Improving robotic cooking using batch bayesian optimization. *IEEE Robotics and Automation Letters* **5**(2), 760–765 (2020)
17. Kellerer, H., Pferschy, U., Pisinger, D.: *Knapsack problems*. Springer (2004)
18. Lai, X., Hao, J., Glover, F.W., Lü, Z.: A two-phase tabu-evolutionary algorithm for the 0-1 multidimensional knapsack problem. *Information Science* **436–437**, 282–301 (2018).
19. Oh, C., Tomczak, J., Gavves, E., Welling, M.: Combinatorial bayesian optimization using the graph cartesian product. In: Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E., Garnett, R. (eds.) *Advances in Neural Information Processing Systems 32*, pp. 2914–2924. Curran Associates, Inc. (2019)
20. Rai, A., Antonova, R., Meier, F., Atkeson, C.G.: Using simulation to improve sample-efficiency of bayesian optimization for bipedal robots. *Journal of Machine Learning Research* **20**(49), 1–24 (2019)

21. Sano, S., Kadowaki, T., Tsuda, K., Kimura, S.: Application of bayesian optimization for pharmaceutical product development. *Journal of Pharmaceutical Innovation* (2019).
22. Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., de Freitas, N.: Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE* **104**(1), 148–175 (2016)
23. Snoek, J., Larochelle, H., Adams, R.P.: Practical bayesian optimization of machine learning algorithms. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems 25*, pp. 2951–2959. Curran Associates, Inc. (2012)
24. Vasquez, M., Vimont, Y.: Improved results on the 0-1 multidimensional knapsack problem. *European Journal of Operational Research* **165**(1), 70–81 (2005)
25. Vimont, Y., Boussier, S., Vasquez, M.: Reduced costs propagation in an efficient implicit enumeration for the 01 multidimensional knapsack problem. *Journal of Combinatorial Optimization* **15**(2), 165–178 (2008)
26. Wang, Z., Hutter, F., Zoghi, M., Matheson, D., De Freitas, N.: Bayesian optimization in a billion dimensions via random embeddings. *J. Artif. Int. Res.* **55**(1), 361–387 (Jan 2016)
27. Wilbaut, C., Hanafi, S.: New convergent heuristics for 0–1 mixed integer programming. *European Journal of Operational Research* **195**(1), 62–74 (2009)
28. Wilbaut, C., Hanafi, S., Salhi, S.: A survey of effective heuristics and their application to a variety of knapsack problems. *IMA Journal of Management Mathematics* **19**(3), 227–244 (2008)
29. Yogatama, D., Kong, L., Smith, N.A.: Bayesian optimization of text representations. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. pp. 2100–2105. Association for Computational Linguistics, Lisbon, Portugal (Sep 2015).