

APPROXIMATION SCHEMES FOR TWO NON-LINEAR KNAPSACK PROBLEMS ARISING IN ALTERNATING CURRENT ELECTRIC SYSTEMS

TRUNG THANH NGUYEN

Hai Phong University; ttnguyen.cs@gmail.com

Abstract. The purpose of this paper is to study the approximability of two non-linear Knapsack problems, which are motivated by important applications in alternating current electrical systems. The first problem is to maximize a nonnegative linear objective function subject to a quadratic constraint, whilst the second problem is a dual version of the first one, where an objective function is minimized. Both problems are NP-hard since they generalize the unbounded Knapsack problem, and it is unlikely to obtain polynomial-time algorithms for them, unless $P = NP$. It is therefore of great interest to know whether or not there exist efficient approximation algorithms which can return an approximate solution in polynomial time with a reasonable approximation factor. Our contribution of this paper is to present polynomial-time approximation schemes (PTASs) and this is the best possible result one can hope for the studied problems. Our algorithm uses a linear programming approach, which is common in the design of approximation schemes.

Keywords. Complex power, alternating current electrical systems, integer quadratic programming, approximation scheme.

1. INTRODUCTION

1.1. Motivation and problem model

The classical Knapsack problem and its variants naturally arise in modelling various practical applications such as computer networks, production planning, health care (see [7] and the references therein), and power allocation (see, e.g., [8, 9, 14]). Motivated by applications in electrical power production and allocation, in this paper we consider two non-linear Knapsack problems. The first problem, called PACKING, is to maximize a nonnegative linear objective function subject to a non-linear and non-separable constraint, and the second one, called COVERING, can be seen as a dual version of the first problem, where an objective is minimized. In what follows, we will give a brief overview of the applications of these two problems as well as their mathematical formulations.

1.1.1. Maximizing the profit of users

Recently, Yu and Chau [17] introduced a non-linear Knapsack problem called *complex-demand knapsack problem*, to capture the power allocation in alternating current (AC) electrical systems, where each electrical device (user) has a power demand expressed as a complex number $d = d^R + id^I$, and there is a limit on the magnitude of total power supply. Since only a limited number of devices can be served and different devices produce different profits when they receive enough power to work, there arises a natural allocation problem: we want to maximize the sum of profits of the chosen

users such that the magnitude of the sum of satisfied demands should not exceed the capacity. For a complex demand d , the real part d^R represents the active power and the imaginary part d^I represents the reactive power. The apparent power $|d| = \sqrt{(d^R)^2 + (d^I)^2}$ is the magnitude of the complex power d . Therefore, the system capacity can be formulated through a quadratic constraint and, in fact, the power allocation can be mathematically modeled as an integer quadratic programming problem. In this paper we consider the complex-demand knapsack problem in a setting where we allow each user to be able to have more than one complex demand. The problem can be formulated as follows.

$$\text{PACKING} \quad \max \sum_{k=1}^n u_k x_k \quad (1)$$

$$\text{subject to} \quad \left(\sum_{k=1}^n d_k^R x_k \right)^2 + \left(\sum_{k=1}^n d_k^I x_k \right)^2 \leq C^2. \quad (2)$$

$$x \in \mathbb{Z}_+^n, \quad (3)$$

where $d_k = (d_k^R, d_k^I) \in \mathbb{Z}_+^2$ is an unit complex demand of power of the k -th user, $C \in \mathbb{Z}_+$ is a real-valued capacity of total satisfiable demand in apparent power, and $u_k \in \mathbb{Z}_+$ is the profit of the k -th user if the unit demand is satisfied.

1.1.2. Minimizing the cost of electrical power generation

The dual version of the PACKING problem above can be used to model the *Unit Commitment Problem* [12] in electrical power production, where the production of a set of electrical generators is coordinated in order to match the energy demand at minimum cost from energy production. There are two common types of cost factors that can be taken into account to measure the cost of electrical power generation, including *capital cost* and *carbon pricing*. Intuitively, the capital costs can be seen as the payment for the purchase of land, buildings, construction, and equipment used in the production of electricity, and this cost may vary depending on the type of power plants. Particularly, it is typically low for fossil fuel power stations, yet high for wind turbines, and become very high for other ways of generating electricity, such as waste to energy, wave and tidal, solar thermal, and nuclear. The carbon pricing is indirectly borne by society as a whole as a consequence of using that energy source. For example, most electricity today is generated by burning fossil fuels and this process results in the transformation of carbon dioxide (CO₂), which is then released into the air. Hence, the carbon pricing could be seen as the cost of fossil fuel generation, which charges those who produce CO₂ from their emissions, and thus is a necessary part of policies that can reduce the global-warming emissions.

Now lets imagine that we have a project that aims to build up a number of different power plants for the generation of electrical power in such a way that the total electrical power generated (i.e., the supply) must be not less than a threshold reflecting the demand on energy consumption, while the total cost induced by the generation is minimized. In order to minimize the total cost of electrical power generation, the decision maker has to decide how many power stations should be constructed, and what type of power stations should be. This problem can be mathematically modeled as an integer non-linear programming, in which one needs to minimize a linear cost function subject to

a quadratic constraint expressing the constraint on the total power generated. The mathematical formulation of the problem is as follows.

$$\text{COVERING} \quad \min \sum_{k=1}^n c_k x_k, \quad (4)$$

$$\text{subject to} \quad \left(\sum_{k=1}^n d_k^R x_k \right)^2 + \left(\sum_{k=1}^n d_k^I x_k \right)^2 \geq C^2, \quad (5)$$

$$x \in \mathbb{Z}_+^n, \quad (6)$$

where n is the number of different types of power plants needed to build. For the power plant of type $k \in \{1, \dots, n\}$, the variable x_k represents the number of alternating current generators that the power plant can have, and each generator can produce an amount of power specified by a complex number $d_k = d_k^R + id_k^I$, and induces a cost c_k . The nonnegative number C represents the energy demand.

1.2. Related work

Although the complex-demand Knapsack problem was introduced recently by Yu and Chau [17] along with an important application in smart grid, it was actually studied for the first time by Woeginger [15, 16] almost twenty years ago, but in a different name, 2-weighted Knapsack problem. In his paper, Woeginger ruled out the existence of fully polynomial time approximation schemes for the problem, unless $P = NP$. Yu and Chau [17] gave a first $(\frac{1}{2} - \epsilon)$ -approximation algorithm for any small constant $\epsilon > 0$. A simple greedy $\frac{1}{2\sqrt{2}}$ -approximation algorithm with running time $O(n \log n)$ was presented recently in [6]. Chau et al. [1, 2] settled the approximability status of the problem by providing a PTAS for the problem, their method makes use of the PTAS for the multi-dimensional Knapsack problem as a subroutine. Very recently, Elbassioni and Nguyen [3, 4] extended the study to more general problems with non-linear objective functions such as fractional functions, submodular functions, and quadratic functions of fixed nonnegative rank.

1.3. Our contribution and technique

PACKING and COVERING belong to the class of integer quadratic programming, and are NP-hard as they generalize the (unbounded) Knapsack problem [7]. This means they cannot be solved in polynomial time, under the standard assumption $P \neq NP$. It is therefore important to look for efficient approximation algorithms with a good guarantee approximation. Our contribution of this paper is to propose the first PTASs for PACKING and COVERING, which are the best possible results we can hope for these problems. Note that the greedy approach in [6], the geometry approach in [2], and the linear programming approach in [4] are for a variant of the PACKING problem with binary variables, and only the last approach can be extended to the setting with integer variable, as shown in our paper, to yield a PTAS with the same time complexity. The theorems below summarize our results.

Theorem 1. *Given a problem instance of PACKING, there is an algorithm that runs in polynomial time in the size of the instance and returns a solution of value of at least $1 - \epsilon$ times the optimum, for any fixed constant $\epsilon > 0$.*

Theorem 2. *Given a problem instance of COVERING, there is an algorithm that runs in polynomial time in the size of the instance and returns a solution of value of at most $1 + \epsilon$ times the optimum, for any fixed constant $\epsilon > 0$.*

1.4. Basic notions on approximation algorithms

For $\alpha \in (0, 1]$, a vector $x \in \mathbb{Z}_+^n$ is said to be an α -approximate solution for problem PACKING (resp., COVERING), if x is a feasible solution satisfying $\sum_{k=1}^n u_k x_k \geq \alpha \cdot \text{OPT}$ (resp., $\sum_{k=1}^n c_k x_k \leq (1/\alpha) \cdot \text{OPT}$), where OPT is the value of an optimal solution. A PTAS for problem PACKING (resp., COVERING) is an algorithm that runs in polynomial time in the input size, for every fixed ϵ , and outputs an $(1 - \epsilon)$ -approximate solution (resp., $(1 + \epsilon)$ -approximate solution). An FPTAS is a PTAS whose running time is also polynomial in $1/\epsilon$. Throughout this paper, we will sometimes write $u^T x$ and $c^T x$ instead of $\sum_{k=1}^n u_k x_k$ and $\sum_{k=1}^n c_k x_k$, respectively, where v^T denotes the transpose of the vector v .

2. APPROXIMATION SCHEME FOR PACKING

In this section we give a proof for Theorem 1. Let $\epsilon > 0$ be a fixed constant, and \mathcal{I} be a problem instance of PACKING, we will show that the Algorithm 1 below runs in polynomial time and outputs a solution z such that $u^T z \geq (1 - \epsilon)\text{OPT}$, where OPT denotes the value of an optimum of \mathcal{I} .

Throughout this paper, to simplify the presentation, we call items instead of users or devices or power plants. Let λ be some constant depending on ϵ and its exact value will be determined later. Without loss of generality, we can assume that $u_1 \geq u_2 \geq \dots \geq u_n$. The idea, extending the one in [5], is to try to guess the λ items of highest utility (profit) in the optimal solution. This can be done by considering all possibilities of choosing a vector $s = (s_1, \dots, s_n) \in \mathbb{Z}_+^n$ such that $\sum_{i=1}^n s_i \leq \lambda$ and

$$\left(\sum_{k=1}^n d_k^R s_k \right)^2 + \left(\sum_{k=1}^n d_k^I s_k \right)^2 \leq C^2. \quad (7)$$

The constraint (7) needs to be satisfied by the vector s since otherwise, s will not be part of the optimum solution. Let \mathcal{X} be the set containing all such vectors s . It is easy to see that the size of \mathcal{X} is bounded by $O(n^\lambda)$, which is polynomial in n for every constant λ . Therefore, we can guess exactly the λ items of highest utility in the optimum in polynomial time, by running over all vectors $s \in \mathcal{X}$. Now, for each $s \in \mathcal{X}$, denote m as the largest positive integer for which $s_m \neq 0$, and let

$$U = \sum_{k=1}^m d_k^R s_k, \quad \text{and} \quad V = \sum_{k=1}^m d_k^I s_k. \quad (8)$$

If $s_i = 0$ for all $i = 1, \dots, n$ then set $m = 0$. We next consider the following problem

$$\text{IP}(s) \quad \max \sum_{k=m}^n u_k x_k, \tag{9}$$

$$\text{subject to} \quad \left(U + \sum_{k=m}^n d_k^R x_k \right)^2 + \left(V + \sum_{k=m}^n d_k^I x_k \right)^2 \leq C^2, \tag{10}$$

$$x \in \mathbb{Z}_+^{n-m+1}. \tag{11}$$

Let us define $\text{CR}(s)$ as a convex relaxation of the $\text{IP}(s)$, which is obtained from $\text{IP}(s)$ by replacing the constraint $x \in \mathbb{Z}_+^{n-m+1}$ by $x \in \mathbb{R}_+^{n-m+1}$. Note that $\text{CR}(s)$ is feasible due to (7). Furthermore, since $\text{CR}(s)$ is convex, its optimal solution y can be found via polynomial-time algorithms for convex programming (see, e.g., [11]).

$$\text{CR}(s) \quad \max \sum_{k=m}^n u_k x_k, \tag{12}$$

$$\text{subject to} \quad \left(U + \sum_{k=m}^n d_k^R x_k \right)^2 + \left(V + \sum_{k=m}^n d_k^I x_k \right)^2 \leq C^2, \tag{13}$$

$$x \in \mathbb{R}_+^{n-m+1}. \tag{14}$$

Algorithm 1 Packing-PTAS

Require: A utility vector $u \in \mathbb{Z}_+^n$; demands $d_k = (d_k^R, d_k^I) \in \mathbb{Z}_+^2$, $k \in \{1, \dots, n\}$; a capacity C ; and accuracy parameter ϵ

Ensure: A solution v such that $u^T v \geq (1 - \epsilon)\text{OPT}$

- 1: Order the x_k 's such that $u_1 \geq \dots \geq u_n$
- 2: $v \leftarrow (0, \dots, 0)$
- 3: $\lambda \leftarrow \lceil \frac{3}{\epsilon} \rceil$
- 4: $\mathcal{X} \leftarrow \{(s_1, \dots, s_n) \in \mathbb{Z}_+^n \mid \sum_{k=1}^n s_k \leq \lambda\}$
- 5: **for** each $s = (s_1, \dots, s_n) \in \mathcal{X}$ **do**
- 6: $m \leftarrow \max\{i \in [n] \mid s_i \neq 0\}$
- 7: Construct $\text{CR}(s)$
- 8: Return y as an optimal solution of $\text{CR}(s)$
- 9: Construct the convex polytope $\mathcal{P}(s)$
- 10: Find a BFS z
- 11: $\bar{x} \leftarrow (s_1, \dots, s_{m-1}, s_m + \lfloor z_m \rfloor, \lfloor z_{m+1} \rfloor, \dots, \lfloor z_n \rfloor)$
- 12: **if** $\sum_{k=1}^n u_k \bar{x}_k > \sum_{k=1}^n u_k v_k$ **then**
- 13: $v \leftarrow \bar{x}$
- 14: **return** v

Define

$$\alpha = \sum_{k=m}^n d_k^R y_k, \quad \text{and} \quad \beta = \sum_{k=m}^n d_k^I y_k, \tag{15}$$

and

$$\gamma = \sum_{k=m}^n (Ud_k^R + Vd_k^I)y_k. \quad (16)$$

Let $\mathcal{P}(s)$ be a (convex) polytope in \mathbb{R}_+^{n-m+1} , defined by the system of following linear constraints

$$\sum_{k=m}^n d_k^R x_k \leq \alpha \quad (17)$$

$$\sum_{k=m}^n d_k^I x_k \leq \beta \quad (18)$$

$$\sum_{k=m}^n (Ud_k^R + Vd_k^I)x_k \leq \gamma \quad (19)$$

$$x_k \geq 0, \text{ for } k = m, \dots, n. \quad (20)$$

Note that $\mathcal{P}(s)$ is a non-empty set as it contains y as a point. Therefore, we can find in polynomial time a basic feasible solution (BFS) z in this polytope such that z has at most 3 nonzero components and that

$$\sum_{k=m}^n u_k z_k \geq \sum_{k=m}^n u_k y_k, \quad (21)$$

(see, e.g., [13, 10]). We then round-down z to obtain an integral solution $(\lfloor z_m \rfloor, \lfloor z_{m+1} \rfloor, \dots, \lfloor z_n \rfloor)$, and this solution is combined with the vector s to result in an integral solution \bar{x} to the problem instance \mathcal{I} :

$$\bar{x} = (s_1, \dots, s_{m-1}, s_m + \lfloor z_m \rfloor, \lfloor z_{m+1} \rfloor, \dots, \lfloor z_n \rfloor).$$

For each vector $s \in \mathcal{X}$, the algorithm outputs a corresponding integral solution \bar{x} . Let v be the solution of maximum value among all such solutions \bar{x} , and return v as a final solution.

Lemma 1. *Algorithm 1 runs in polynomial time and produces a solution whose value is at least a fraction of $(1 - \epsilon)$ of the optimum.*

Proof. From the argument above, it can be easily seen that the running time of the Algorithm 1 is polynomial in the size of the input, for any fixed constant ϵ . Let $x^* = (x_1^*, \dots, x_n^*)$ be an optimal solution to the instance \mathcal{I} , and let $\text{OPT} = u^T x^*$. We now argue that the solution v returned by the algorithm is such that $u^T v \geq (1 - \epsilon)\text{OPT}$. If $\lambda \geq \sum_{k=1}^n x_k^*$, then v is an exact optimal solution to the

\mathcal{I} (this is because of the guessing step). Suppose that $\lambda < \sum_{k=1}^n x_k^*$. Let $s = (s_1, \dots, s_m, \dots, s_n)$ be the vector such that

$$i) \sum_{k=1}^n s_k = \lambda,$$

$$ii) s_m \neq 0 \text{ and } s_k = 0 \text{ for all } k = m + 1, \dots, n,$$

$$iii) s_k = x_k^* \text{ for all } k = 1, \dots, m - 1.$$

Clearly, $s \in \mathcal{X}$. Then, the feasibility of x^* for $\text{CR}(s)$ guarantees that (7) holds, and hence an optimal solution y to $\text{CR}(s)$ is found in Step 8 of the algorithm. Furthermore, the feasibility of y for

CR(s) implies that

$$\begin{aligned} \left(U + \sum_{k=m}^n d_k^R y_k \right)^2 + \left(V + \sum_{k=m}^n d_k^I y_k \right)^2 &= U^2 + V^2 + \left(\sum_{k=m}^n d_k^I y_k \right)^2 + \left(\sum_{k=m}^n d_k^R y_k \right)^2 \\ &\quad + 2U \cdot \left(\sum_{k=m}^n d_k^I y_k \right) + 2V \cdot \left(\sum_{k=m}^n d_k^R y_k \right) \leq C^2, \end{aligned}$$

or,

$$\alpha^2 + \beta^2 + 2\gamma \leq C^2 - U^2 - V^2, \quad (22)$$

where $\alpha, \beta, \gamma, U, V$ are defined as in (8), (15), and (16). Moreover, the definition of α, β, γ implies that $y \in \mathcal{P}(s)$. This means $\mathcal{P}(s)$ is nonempty and thus, there must be a BFS z of $\mathcal{P}(s)$ such that $\sum_{k=m}^n u_k z_k \geq \sum_{k=m}^n u_k y_k$. Let $\bar{x} \in \mathbb{Z}_+^n$ be the rounded solution obtained from z in Step 11. We have

$$\sum_{k=1}^n u_k v_k \geq \sum_{k=1}^n u_k \bar{x}_k = \sum_{k=1}^{m-1} u_k \bar{x}_k + \sum_{k=m}^n u_k \bar{x}_k. \quad (23)$$

Since $(\bar{x}_m, \bar{x}_{m+1}, \dots, \bar{x}_n)$ is obtained by rounding down z and the fact that z has at most three nonzero components, it follows that

$$\sum_{k=m}^n u_k \bar{x}_k \geq u_m s_m + \sum_{k=m}^n u_k z_k - 3u_m \quad (24)$$

Note that $u_1 \geq \dots \geq u_n$, it follows that

$$3u_m = 3u_m \cdot \frac{1}{\lambda} \sum_{k=1}^m s_k \leq \frac{3}{\lambda} \sum_{k=1}^m u_k s_k, \quad (25)$$

since

$$\lambda = \sum_{k=1}^n s_k = \sum_{k=1}^m s_k$$

By choosing $\lambda = \left\lceil \frac{3}{\epsilon} \right\rceil$, it follows that $\frac{3}{\lambda} \leq \epsilon$, and that

$$3u_m \leq \epsilon \sum_{k=1}^{m-1} u_k \bar{x}_k + \epsilon u_m s_m, \quad (26)$$

From (23), (24), and (26) we have

$$\begin{aligned}
\sum_{k=1}^n u_k v_k &\geq \sum_{k=1}^n u_k \bar{x}_k \\
&\geq \sum_{k=1}^{m-1} u_k \bar{x}_k + u_m s_m - \epsilon \sum_{k=1}^{m-1} u_k \bar{x}_k - \epsilon u_m s_m + \sum_{k=m}^n u_k z_k \\
&\geq (1 - \epsilon) \left(\sum_{k=1}^{m-1} u_k \bar{x}_k + u_m s_m + \sum_{k=m}^n u_k z_k \right) \\
&\geq (1 - \epsilon) \left(\sum_{k=1}^{m-1} u_k \bar{x}_k + u_m s_m + \sum_{k=m}^n u_k y_k \right) \\
&\geq (1 - \epsilon) \left(\sum_{k=1}^{m-1} u_k \bar{x}_k + \sum_{k=m}^n u_k x_m^* \right) \\
&= (1 - \epsilon) \text{OPT}.
\end{aligned}$$

It remains to argue that \bar{x} (and thus v) is feasible for PACKING. Let

$$T = \left(\sum_{k=1}^n d_k^R \bar{x}_k \right)^2 + \left(\sum_{k=1}^n d_k^I \bar{x}_k \right)^2,$$

we prove that $T \leq C^2$. Indeed, we transform T as follows:

$$\begin{aligned}
T &= \left(\sum_{k=1}^m d_k^R s_k \right)^2 + \left(\sum_{k=1}^m d_k^I s_k \right)^2 + \left(\sum_{k=m+1}^n d_k^R \bar{x}_k + d_m^R \lfloor z_m \rfloor \right)^2 + \left(\sum_{k=m+1}^n d_k^I \bar{x}_k + d_m^I \lfloor z_m \rfloor \right)^2 \\
&\quad + 2 \left(\sum_{k=1}^m d_k^R s_k \right) \cdot \left(\sum_{k=m+1}^n d_k^R \bar{x}_k + d_m^R \lfloor z_m \rfloor \right) + 2 \left(\sum_{k=1}^m d_k^I s_k \right) \cdot \left(\sum_{k=m+1}^n d_k^I \bar{x}_k + d_m^I \lfloor z_m \rfloor \right).
\end{aligned}$$

Since $z \in \mathcal{P}(s)$, and $\bar{x}_k \leq z_k$ for all $k \geq m+1$, it follows that

$$\sum_{k=m+1}^n d_k^R \bar{x}_k + d_m^R \lfloor z_m \rfloor \leq \sum_{k=m}^n d_k^R z_k \leq \alpha,$$

and

$$\sum_{k=m+1}^n d_k^I \bar{x}_k + d_m^I \lfloor z_m \rfloor \leq \sum_{k=m}^n d_k^I z_k \leq \beta.$$

Therefore,

$$T \leq U^2 + V^2 + \alpha^2 + \beta^2 + 2\gamma \leq C^2.$$

The last inequality follows from (22). This completes the proof. \blacksquare

3. APPROXIMATION SCHEME FOR COVERING

This section is devoted to presenting a proof of Theorem 2, which claims the existence of a polynomial time approximation scheme for the COVERING problem. Basically, the main idea for designing such a scheme is not very different from the one for the PACKING problem in the previous section, which relies on the polynomial time solvability of the convex relaxation CR(s). However, a similar relaxation (see below) for COVERING is no longer convex due to the non-convexity of the quadratic constraint. Consequently, the current techniques known for solving exactly convex problems in polynomial time cannot be applied to the case of COVERING.

$$(NCR) \quad \max \sum_{k=1}^n c_k x_k, \tag{27}$$

$$\text{subject to} \quad \left(\sum_{k=1}^n d_k^R x_k \right)^2 + \left(\sum_{k=1}^n d_k^I x_k \right)^2 \geq C^2, \tag{28}$$

$$x \in \mathbb{R}_+^n, \tag{29}$$

Fortunately, by using a the KKT conditions, one can show that the global optimum to the non-convex relaxation NCR above can be found in polynomial time.

Theorem 3. *One can find an optimal solution z to the problem NCR in polynomial time. Furthermore, z has at most two non-zero components.*

Proof. To simplify the notations, let $d_k^R = a_k$ and $d_k^I = b_k$ for all $k = 1, \dots, n$.

Without of loss generality we can assume that there is no pair of indices (i, j) such that

$$\frac{a_i}{a_j} = \frac{b_i}{b_j} = \frac{c_i}{c_j}. \tag{30}$$

The argument is as follows. Suppose that the condition (30) holds for some $i \neq j$. Let $a_i = \ell a_j$, $b_i = \ell b_j$, and $c_i = \ell c_j$ for some positive integer ℓ . By reordering the variables (if necessary) we can assume that $i = n - 1, j = n$. Hence, the objective function of NCR and the constraint (28) can be rewritten, respectively, in the following forms:

$$\sum_{k=1}^{n-2} c_k x_k + c_n y,$$

and,

$$\left(\sum_{k=1}^{n-2} a_k x_k + a_n y \right)^2 + \left(\sum_{k=1}^{n-2} b_k x_k + b_n y \right)^2 \geq C^2,$$

where $y = \ell x_{n-1} + x_n$. This means that the problem NCR is reduced to a similar problem, but with a smaller number of variables. By repeating the same reduction until the condition (30) is not satisfied for all $i \neq j$, we obtain a reduced problem which are equivalent to NCR, and we can therefore solve this problem instead of NCR.

We are now ready to prove the Theorem 3. Let us define the Lagrangian function

$$\mathcal{L}(x, \lambda, \mu) = \sum_{k=1}^n c_k x_k - \sum_{k=1}^n \mu_k x_k + \lambda \left\{ C^2 - \left(\sum_{k=1}^n a_k x_k \right)^2 - \left(\sum_{k=1}^n b_k x_k \right)^2 \right\},$$

where $\lambda, \mu = (\mu_1, \dots, \mu_n)$ are called Lagrange multipliers, and $x = (x_1, \dots, x_n)$.

It is known that every local solution to NCR must satisfy the following Karush-Kuhn-Tucker (KKT) conditions

$$\frac{\partial \mathcal{L}(x, \lambda, \mu)}{\partial x_k} = 0, \text{ for } k = 1, \dots, n, \quad (31)$$

$$\lambda \left\{ C^2 - \left(\sum_{k=1}^n a_k x_k \right)^2 - \left(\sum_{k=1}^n b_k x_k \right)^2 \right\} = 0, \quad (32)$$

$$C^2 - \left(\sum_{k=1}^n a_k x_k \right)^2 - \left(\sum_{k=1}^n b_k x_k \right)^2 \leq 0, \quad (33)$$

$$\lambda \geq 0, \quad (34)$$

$$\mu_k x_k = 0, \text{ } k = 1, \dots, n, \quad (35)$$

$$\mu_k \geq 0, \text{ } k = 1, \dots, n, \quad (36)$$

$$x_k \geq 0, \text{ } k = 1, \dots, n. \quad (37)$$

A feasible solution to NCR which satisfies the system of equations above is called a KKT point (or a stationary point). Since every optimal point needs to be among the KKT points, in order to find an optimal point, we just need to maximize the objective function over all possible KKT points. This task in general is impossible because of the two reasons: first, the number of KKT points could be exponential; and second, determining KKT points might be very hard. In what follows, however, we will show that there is a polynomial number of KKT points, which can be determined efficiently, for the system of equations (31)-(37) and thus, the optimal point can be found easily. The equations (31) is equivalent to

$$c_k - 2\lambda \left(a_k \sum_{k=1}^n a_k x_k + b_k \sum_{k=1}^n b_k x_k \right) - \mu_k = 0, \quad (38)$$

for $k = 1, \dots, n$.

Let

$$X = 2\lambda \sum_{k=1}^n a_k x_k \quad \text{and} \quad Y = 2\lambda \sum_{k=1}^n b_k x_k. \quad (39)$$

Then the equations (38) can be written as

$$a_k X + b_k Y + \mu_k = c_k, \quad (40)$$

for $k = 1, \dots, n$.

Now if $\lambda = 0$, it follows that $\mu_k = c_k \neq 0$ due to (31), and by (35) $x_k = 0$ for all k , but this violates the constraint (33). Hence, $\lambda > 0$, and there must be at least one variable of positive value.

Furthermore, the constraint (32) implies that

$$C^2 = \left(\sum_{k=1}^n a_k x_k \right)^2 + \left(\sum_{k=1}^n b_k x_k \right)^2 = \left(\frac{X}{2\lambda} \right)^2 + \left(\frac{Y}{2\lambda} \right)^2. \tag{41}$$

We now can find a set S of all KKT points as follows. It is sufficient to consider the following cases:

Case I: All variables μ_k are non-zero.

Case II: There is exactly one zero variable μ_k .

Case III: There are at least two zero variables μ_i, μ_j .

In the case I, it follows from (35) that $x_k = 0$ for all $k = 1, \dots, n$, but then the zero vector $(0, \dots, 0)$ is not feasible to NCR. For the case II, there are in total n possibilities for choosing a zero variable μ_k . In fact, for $k \in \{1, \dots, n\}$, the k th possibility is that $\mu_k = 0$ and $\mu_i \neq 0$ for all $i \neq k$. Again, this follows from (35) that $x_i = 0$ for all $i \neq k$, and we thus need to check if a vector of the form $(0, \dots, 0, x_k, 0, \dots, 0)$ is a solution of the system (31)-(37). If yes, we add it, with the determined value of x_k , into the set S .

For the case III, there are in total $\binom{n}{2}$ possibilities of choosing two zero variables $(\mu_i, \mu_j), i \neq j$. For each such a possibility (μ_i, μ_j) , the inequalities (40) result in the following equations

$$a_i X + b_i Y = c_i, \tag{42}$$

$$a_j X + b_j Y = c_j. \tag{43}$$

Note that by our early assumption (which says that there is no pair (i, j) satisfying (30)) it holds that the system (42)-(43) either has no solution or has only one solution of the form

$$X = \frac{c_i b_j - c_j b_i}{a_i b_j - a_j b_i}, \quad Y = \frac{a_i c_j - a_j c_i}{a_i b_j - a_j b_i}. \tag{44}$$

Substituting (44) into (40) and (41) we can easily find the values of λ and μ_k for all k . Additionally, the values of $\sum_{k=1}^n a_k x_k$ and $\sum_{k=1}^n b_k x_k$ can be determined by the equalities (38). We further check if these values satisfy all other KKT conditions and if yes, we solve the following problem

$$\min \sum_{k=1}^n c_k x_k, \tag{45}$$

$$\text{subject to } \sum_{k=1}^n a_k x_k = X/2\lambda, \tag{46}$$

$$\sum_{k=1}^n b_k x_k = Y/2\lambda, \tag{47}$$

$$x \in \mathbb{R}_+^n, \tag{48}$$

$$x_k = 0 \quad \text{if } \mu_k \neq 0, \tag{49}$$

which, in fact, can be thus solved in polynomial time. The optimal solution found is then put into S . Note that this optimal solution has at most two non-zero components since there are only two non-trivial linear constraints (46) and (47).

Finally, the optimal solution z to the problem NCR is computed as

$$z = \arg_{x \in S} \sum_{k=1}^n c_k x_k.$$

Note that since the size of S is at most $n + \binom{n}{2} = O(n^2)$, z can be computed in polynomial time. Furthermore, it is observed that z also has at most two non-zero components. This completes the proof of Theorem 3. \blacksquare

Turning to the proof of Theorem 2, let us consider the following Algorithm 2.

Algorithm 2 Covering-PTAS

Require: A cost vector $c \in \mathbb{Z}_+^n$; demands $d_k = (d_k^R, d_k^I) \in \mathbb{Z}_+^2$, $k \in \{1, \dots, n\}$; a capacity C ; and accuracy parameter ϵ

Ensure: A solution v such that $c^T v \leq (1 + \epsilon)\text{OPT}$

- 1: Order the x_k 's such that $c_1 \geq \dots \geq c_n$
 - 2: $v \leftarrow (1, \dots, 1)$
 - 3: $\lambda \leftarrow \lceil \frac{2}{\epsilon} \rceil$
 - 4: $\mathcal{X} \leftarrow \{(s_1, \dots, s_n) \in \mathbb{Z}_+^n \mid \sum_{k=1}^n s_k \leq \lambda\}$
 - 5: **for** each $s = (s_1, \dots, s_n) \in \mathcal{X}$ **do**
 - 6: $m \leftarrow \max\{i \in [n] \mid s_i \neq 0\}$
 - 7: Construct $\text{NCR}(s)$
 - 8: Return z as an optimal solution of $\text{CR}(s)$, which has at most two non-zero components
 - 9: $\bar{x} \leftarrow (s_1, \dots, s_{m-1}, s_m + \lceil z_m \rceil, \lceil z_{m+1} \rceil, \dots, \lceil z_n \rceil)$
 - 10: **if** $\sum_{k=1}^n c_k \bar{x}_k < \sum_{k=1}^n c_k v_k$ **then**
 - 11: $v \leftarrow \bar{x}$
 - 12: **return** v
-

Lemma 2. *For any fixed $\epsilon > 0$, Algorithm 2 runs in polynomial time and produces a solution whose value is at most a fraction of $1 + \epsilon$ of the optimum.*

Proof. The proof is essentially the same as the one given for Algorithm 1. Let $\lambda = \lceil \frac{2}{\epsilon} \rceil$. The quantities U, V with respect to a vector $s \in \mathcal{X}$ are defined in a same way as before. The non-convex relaxation $\text{NCR}(s)$ is

$$\text{NCR}(s) \quad \max \sum_{k=m}^n c_k x_k, \tag{50}$$

$$\text{subject to} \quad \left(U + \sum_{k=m}^n d_k^R x_k \right)^2 + \left(V + \sum_{k=m}^n d_k^I x_k \right)^2 \geq C^2, \tag{51}$$

$$x \in \mathbb{R}_+^{n-m+1}. \tag{52}$$

Let $x^* = (x_1^*, \dots, x_n^*)$ be an optimal solution to the instance \mathcal{I} of COVERING, and let $\text{OPT} = c^T x^*$. We now argue that the solution v returned by the algorithm is such that $c^T v \leq (1 + \epsilon)\text{OPT}$. If $\lambda \geq \sum_{k=1}^n x_k^*$, then we are done. Suppose that $\lambda < \sum_{k=1}^n x_k^*$. Let $s = (s_1, \dots, s_m, \dots, s_n)$ be the vector such that

- i) $\sum_{k=1}^n s_k = \lambda$,
- ii) $s_m \neq 0$ and $s_k = 0$ for all $k = m + 1, \dots, n$,
- iii) $s_k = x_k^*$ for all $k = 1, \dots, m - 1$.

We have that

$$\sum_{k=1}^n c_k v_k \leq \sum_{k=1}^n c_k \bar{x}_k = \sum_{k=1}^{m-1} c_k \bar{x}_k + \sum_{k=m}^n c_k \bar{x}_k. \tag{53}$$

Note that this time $(\bar{x}_m, \bar{x}_{m+1}, \dots, \bar{x}_n)$ is obtained by rounding up z , and by the fact that z has at most two non-zero components, it follows that

$$\sum_{k=m}^n c_k \bar{x}_k \leq c_m s_m + \sum_{k=m}^n c_k z_k + 2c_m. \tag{54}$$

Note that $c_1 \geq \dots \geq c_n$, it follows that

$$2u_m = 2c_m \cdot \frac{1}{\lambda} \sum_{k=1}^m s_k \leq \frac{2}{\lambda} \sum_{k=1}^m c_k s_k = \epsilon \sum_{k=1}^{m-1} c_k \bar{x}_k + \epsilon c_m s_m, \tag{55}$$

since

$$\lambda = \sum_{k=1}^n s_k = \sum_{k=1}^m s_k,$$

From (53), (54), and (55) we have

$$\begin{aligned} \sum_{k=1}^n c_k v_k &\leq \sum_{k=1}^n c_k \bar{x}_k \\ &\leq \sum_{k=1}^{m-1} c_k \bar{x}_k + c_m s_m + \epsilon \sum_{k=1}^{m-1} c_k \bar{x}_k + \epsilon c_m s_m + \sum_{k=m}^n c_k z_k \\ &\leq (1 + \epsilon) \left(\sum_{k=1}^{m-1} c_k \bar{x}_k + c_m s_m + \sum_{k=m}^n c_k z_k \right) \\ &\leq (1 + \epsilon) \left(\sum_{k=1}^{m-1} c_k \bar{x}_k + \sum_{k=m}^n c_k x_k^* \right) \\ &= (1 + \epsilon) \text{OPT}. \end{aligned}$$

The feasibility of \bar{x} (and v) to the COVERING problem can be treated in a similar way as in the case of PACKING. This completes the proof. ■

4. CONCLUSION

In this paper we have presented the first PTASs for the two non-linear Knapsack problems, PACKING and COVERING, which are motivated by applications in alternating current electrical systems. Since these two problems are NP-hard, the obtained PTASs are the best possible approximation

algorithms, assuming that $P \neq NP$. Although the PTAS is of theoretical interest, it is quite impractical as the running time is exponential in the constant $\frac{1}{\epsilon}$. Our work raises several open problems for the future work. The first problem is to check whether we can achieve a PTAS for COVERING with binary variables. The technique used in this paper, which relies on the KKT conditions, does not help to solve the (non-convex) relaxation of the problem in the binary setting. Another open problem is to extend our PTASs to the case where the constraint involves more than two squares of linear functions. Finally, it would be very interesting to develop more practical approximation algorithms for both PACKING and COVERING. For example, it is worth seeing if the greedy approach provided in [6] yields an approximation algorithm with the same approximation factor of $\frac{1}{2\sqrt{2}}$.

Acknowledgments

We would like to thank Khaled Elbassioni and Gerhard Woeginger for helpful discussion on the problem. Also, we would like to thank the reviewers for all the very helpful comments, which helped to much improve the presentation of the paper. This work was supported by Vietnam National Foundation for Science and Technology Development (NAFOSTED Project No. 102.01-2015.33).

REFERENCES

- [1] C. Chau, K. M. Elbassioni, and M. Khonji, “Truthful mechanisms for combinatorial allocation of electric power in alternating current electric systems for smart grid,” *ACM Trans. Economics and Comput.*, vol. 5, no. 1, p. 7, 2016.
- [2] S. C. Chau, K. M. Elbassioni, and M. Khonji, “Truthful mechanisms for combinatorial AC electric power allocation,” in *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14, Paris, France, May 5-9, 2014*, 2014, pp. 1005–1012.
- [3] K. M. Elbassioni and T. T. Nguyen, “Approximation schemes for multi-objective optimization with quadratic constraints of fixed cp-rank,” in *Algorithmic Decision Theory - 4th International Conference, ADT 2015, Lexington, KY, USA, September 27-30, 2015, Proceedings*, 2015, pp. 273–287.
- [4] —, “Approximation algorithms for binary packing problems with quadratic constraints of low cp-rank decompositions,” *Discrete Applied Mathematics*, vol. 230, pp. 56–70, 2017.
- [5] A. Frieze and M. Clarke, “Approximation algorithms for the m -dimensional 0-1 knapsack problem: worst-case and probabilistic analyses,” *Eur. J. Oper. Res.*, vol. 15, pp. 100–109, 1984.
- [6] A. Karapetyan, M. Khonji, C. Chau, K. M. Elbassioni, and H. H. Zeineldin, “Efficient algorithm for scalable event-based demand response management in microgrids,” *IEEE Transactions on Smart Grid*, doi: 10.1109/TSG.2016.2616945. To appear.
- [7] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*. Springer, Berlin, Germany, 2004.
- [8] N. Kumaraguruparan, H. Sivaramakrishnan, and S. S. Sapatnekar, “Residential task scheduling under dynamic pricing using the multiple knapsack method,” in *IEEE PES Innovative Smart Grid Technologies Conference, ISGT 2012, Washington, DC, USA, January 16-20, 2012*, 2012, pp. 1–6.

- [9] N. Morimoto, Y. Fujita, M. Yoshida, H. Yoshimizu, M. Takiyamada, T. Akehi, and M. Tanaka, "A power allocation management system using an algorithm for the knapsack problem," in *IEEE 38th Annual Computer Software and Applications Conference, COMPSAC Workshops 2014, Vasteras, Sweden, July 21-25, 2014*, 2014, pp. 590–595.
- [10] G. L. Nemhauser and L. A. Wolsey, *Integer and Combinatorial Optimization*. New York, NY, USA: Wiley-Interscience, 1999.
- [11] A. Nemirovski and M. Todd, "Interior-point methods for optimization," *Acta Numerica*, vol. 17, pp. 191–234, 5 2008.
- [12] N. Padhy, "Unit commitment – a bibliographical survey," *IEEE Transaction On Power Systems*, vol. 19, no. 2, pp. 1196–1205, 2004.
- [13] A. Schrijver, *Theory of Linear and Integer Programming*. New York: Wiley, 1986.
- [14] O. A. Sianaki, O. Hussain, and A. R. Tabesh, "A knapsack problem approach for achieving efficient energy consumption in smart grid for endusers' life style," in *2010 IEEE Conference on Innovative Technologies for an Efficient and Reliable Electricity Supply (CITRES)*, 2010, pp. 159–164.
- [15] G. J. Woeginger, "When does a dynamic programming formulation guarantee the existence of an fptas?" in *Proceedings of the Tenth Annual ACM-SIAM Symposium on Discrete Algorithms, 17-19 January 1999, Baltimore, Maryland.*, 1999, pp. 820–829.
- [16] —, "When does a dynamic programming formulation guarantee the existence of a fully polynomial time approximation scheme (FPTAS)?" *INFORMS Journal on Computing*, vol. 12, no. 1, pp. 57–74, 2000.
- [17] L. Yu and C. Chau, "Complex-demand knapsack problems and incentives in AC power systems," in *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13, Saint Paul, MN, USA, May 6-10, 2013*, 2013, pp. 973–980.

Received on September 14, 2017

Revised on November 10, 2017