ELSEVIER

Operations Research Letters 30 (2002) 159–166

Operations Research Letters

www.elsevier.com/locate/dsw

The quadratic 0–1 knapsack problem with series–parallel support

David J. Rader Jr.[a,*], Gerhard J. Woeginger[b]

[a] *Department of Mathematics, Rose-Hulman Institute of Technology, Terre Haute, IN 47803, USA*
[b] *Faculty of Mathematics, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands*

Received 6 March 2002; received in revised form 11 March 2002; accepted 14 March 2002

Abstract

We consider various special cases of the quadratic 0–1 knapsack problem (QKP) for which the underlying graph structure is fairly simple. For the variant with edge series–parallel graphs, we give a dynamic programming algorithm with pseudo-polynomial time complexity, and a fully polynomial time approximation scheme. In strong contrast to this, the variant with vertex series–parallel graphs is shown to be strongly NP-complete.

Keywords: Knapsack problem; Dynamic programming; Computational complexity; Approximability; Approximation scheme

1. Introduction

The *quadratic* 0–1 *knapsack problem* (*QKP*) was introduced by Gallo et al. [7] as a generalization of the classical knapsack problem. It is formulated as follows:

$$\begin{aligned} \text{maximize } & F(\mathbf{x}) = \sum_{i\in V} c_i x_i + \sum_{(i,j)\in E} c_{ij} x_i x_j \\ \text{subject to } & \sum_{i\in V} a_i x_i \leqslant b \\ & \mathbf{x} \in \{0,1\}^n, \end{aligned}$$

where $V = \{1, 2, \ldots, n\}$ is the set of variables and E is the set of quadratic terms. The a_i are called *weight* coefficients, and the c_i and c_{ij} are called *cost* coefficients. All coefficients a_i, c_i and c_{ij} are integral. Throughout this paper, we shall assume that $b \geqslant 0$ and that $0 \leqslant a_i \leqslant b$ for all $i \in V$. Furthermore, we shall assume that $a_i + a_j \leqslant b$ for all $(i,j) \in E$. Note that we do not make any a priori assumptions on the signs of the cost coefficients c_i and c_{ij}. The special case of the QKP where $E = \emptyset$ is the classical *knapsack problem*. Hence, QKP is an NP-hard problem. The special case of the QKP where $c_{ij} \geqslant 0$ holds for all quadratic coefficients is the so-called *supermodular knapsack problem*.

QKP has been used to model selection problems in telecommunications [19] as well as compiler design problems [13]. Gallo et al. [7] give a branch and bound algorithm for QKP that is based on linear upper bound functions known as upper planes. For the supermodular knapsack problem, the Lagrangian relaxation can be solved efficiently and yields quick and easy upper bounds. Branch and bound algorithms that exploit these bounds have been developed by Billionnet et al. [3], Chaillou et al. [5], Gallo and Simeone [8], Hammer and Rader [10], Michelon and Veilleux [15]. Another Lagrangian relaxation that is solvable through a number of continuous linear knapsack

* Corresponding author.
E-mail addresses: david.rader@rose-hulman.edu (D.J. Rader Jr.), g.j.woeginger@math.utwente.nl (G.J. Woeginger).

0167-6377/02/$ - see front matter
PII: S0167-6377(02)00122-0

problems has recently been proposed by Caprara et al. [4]. A few additional ways to solve the general quadratic 0–1 knapsack problem, especially when some of the c_{ij} are negative, are LP relaxations [2] and semidefinite relaxations [11]. For more information on the QKP, we refer the reader to Rader [16].

Until recently, most work on QKP has concentrated on the supermodular case. In addition, very little work has dealt with exploiting the graph–theoretic structure of QKPs. Billionnet [1] has shown that the cardinality-constrained quadratic 0–1 knapsack problem can be solved in $O(n)$ steps if the graph $G=(V,E)$ induced by the objective function is a tree. In this paper, we will derive pseudo-polynomial time algorithms for the special case of QKP where the underlying graph G is *edge series–parallel*. The class of edge series–parallel graphs contains all trees. In contrast to this, QKP with an underlying *vertex series–parallel* graph is strongly NP-hard and cannot have a pseudo-polynomial time solution unless P = NP.

An approach that has not yet been examined for QKPs are fully polynomial time approximation schemes (FPTAS). An FPTAS is an approximation algorithm that, for any given $\varepsilon > 0$, finds a feasible solution with objective value within a factor of $(1-\varepsilon)$ of the optimal objective value. The running time of an FPTAS is polynomially bounded in the input size and in $1/\varepsilon$. An FPTAS is the strongest possible polynomial time approximation result that can be derived for an NP-hard problem (unless P = NP). Ibarra and Kim [12] and Lawler [14] gave such an FPTAS for the classical knapsack problem. In this paper, we design an FPTAS for the special case of QKP where the underlying graph G is edge series–parallel and where all cost coefficients are non-negative.

Organization of the paper. Section 2 recalls all necessary definitions and facts around edge and vertex series–parallel graphs as will be needed in the rest of this paper. Section 3 describes two dynamic programming algorithms for the QKP on edge series–parallel graphs. The time complexity of one of these algorithms is pseudo-polynomially bounded in the weights, and the time complexity of the other algorithm is pseudo-polynomially bounded in the costs. Section 4 translates the second dynamic programming algorithm into an FPTAS. Section 5 proves the NP-hardness result for vertex series–parallel graphs, and Section 6 contains the conclusion.

2. Series–parallel graphs

In this section, we provide definitions of edge series–parallel (ESP) graphs and of vertex series–parallel (VSP) graphs.

A standard way of defining edge series–parallel graphs is via so-called 2-terminal graphs; cf. Duffin [6] and Takamizawa et al. [17]. A 2-terminal graph $G=(V,E)$ is a graph with two special vertices that are called the left terminal and the right terminal. For two 2-terminal graphs $G_i=(V_i,E_i)$, $i=1,2$, we define the following operations:

- The series composition $G_s = G_1 * G_2$ of G_1 and G_2 results from identifying the right terminal of G_1 with the left terminal of G_2. The obtained graph G_s is regarded as a 2-terminal graph whose left terminal is the left terminal of G_1, and whose right terminal is the right terminal of G_2.
- The parallel composition $G_p = G_1 \| G_2$ of G_1 and G_2 results from identifying both right terminals with each other and both left terminals with each other. The terminal vertices of G_p are these identified terminals.

The class of ESP graphs is now defined by the following three rules: (1) The graph consisting of two terminals connected by a single edge is ESP. (2) If G_1 and G_2 are ESP, then $G_1 * G_2$ and $G_1 \| G_2$ are ESP. (3) No other graphs than those defined by (1) and (2) are ESP.

An ESP graph can be decomposed into its atomic parts according to the series and parallel operations in linear time (see [18]). Essentially, such a decomposition corresponds to a binary tree where all interior vertices are labeled by s (series composition) or p (parallel composition), and where all leaves correspond to edges of the graph. See Fig. 1 for an illustration. Note that there can be different binary tree decompositions for a given edge series–parallel graph. We associate with every interior vertex v of a decomposition tree the ESP graph $G(v)$ that is induced by the leaves of the subtree below v. It is well-known that a simple ESP graph $G=(V,E)$ satisfies $|E| \leqslant 2|V|-1$.

Now let us turn to vertex series–parallel graphs. It is convenient to introduce VSP graphs via so-called *minimal vertex series–parallel* digraphs (MVSP for short). See Valdes et al. [18] for more information.

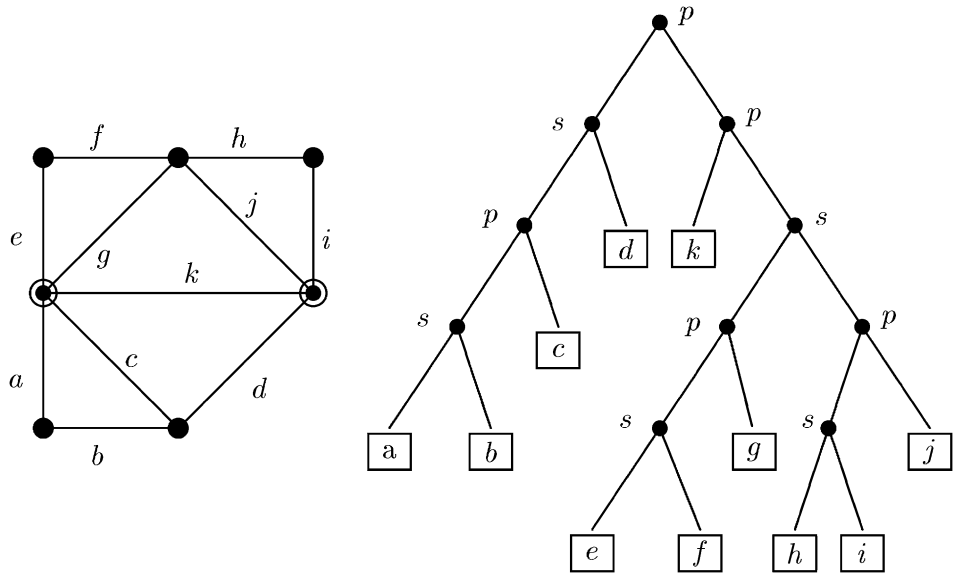

Fig. 1. An edge series–parallel graph and its binary decomposition tree.

For two digraphs $G_i = (V_i, E_i)$, $i = 1, 2$, we define the following operations:

- The series composition of G_1 and G_2 has vertex set $V_1 \cup V_2$ and arc set $E_1 \cup E_2 \cup (T_1 \times S_2)$ where T_1 is the set of sinks in G_1 and S_2 is the set of sources in G_2.
- The parallel composition of G_1 and G_2 has vertex set $V_1 \cup V_2$ and arc set $E_1 \cup E_2$.

The class of MVSP digraphs is defined by the following three rules: (1) The graph consisting of a single vertex is MVSP. (2) If digraphs G_1 and G_2 are MVSP, then their series composition and their parallel composition is also MVSP. (3) No other digraphs than those defined by (1) and (2) are MVSP. Finally, an undirected graph is VSP if it is the underlying undirected graph of a digraph whose transitive closure equals the transitive closure of some MVSP.

Note that every complete bipartite graph $K_{m,n}$ is VSP. In the rest of this paper, VSP graphs will only show up in Theorem 5.1.

3. Dynamic programming algorithms

In this section we develop two dynamic programming algorithms for the special case of QKP where the underlying graph $G = (V, E)$ is ESP. The first algorithm is pseudo-polynomial in the weights, and the second algorithm is pseudo-polynomial in the costs.

3.1. The first DP algorithm

In this subsection, we design an algorithm that is pseudo-polynomial in the weights, and polynomial in the costs.

Consider an instance of QKP where the underlying graph $G = (V, E)$ is ESP. As a first step, we apply the method of Valdes et al. [18] to compute a binary decomposition tree for G in linear time. Consider some interior vertex v in this binary decomposition tree. We denote by $G(v)$ the ESP graph that is induced by the edges in the leaves of the subtree below v. Let $\ell(v)$ be the left terminal and $r(v)$ be the right terminal of $G(v)$. The restriction of QKP to vertices in $G(v)$ is denoted by QKP(v).

Definition 3.1. For a vertex v in some fixed binary decomposition tree, for $\ell, r \in \{0, 1\}$, and for an integer A with $0 \leqslant A \leqslant b$, we denote by $f[v; \ell, r; A]$ the maximum possible objective value of QKP(v) over all feasible solutions that satisfy the conditions

- $x_{\ell(v)} = \ell$ and $x_{r(v)} = r$, and

- $\sum_i a_i x_i = A$ where the summation is done over all the vertices i in $G(v)$.

If there is no feasible solution of this form, then $f[v;\ell,r;A] = -\infty$.

We compute all these values $f[v;\ell,r;A]$ by a dynamic programming approach that starts in the leaves of the given binary decomposition tree, and then moves upwards towards the root. If v is a leaf, then we set

$$f[v;\ell,r;a_{\ell(v)}\ell + a_{r(v)}r] = c_{\ell(v)}\ell + c_{r(v)}r + c_{\ell(v),r(v)}\ell r$$

and we set $f[v;\ell,r;A] = -\infty$ whenever $A \neq a_{\ell(v)}\ell + a_{r(v)}r$. In the remaining cases, vertex v is an interior vertex. If v is a p vertex with left child u and right child w, then we set

$$\begin{aligned} f[v;\ell,r;A] := \max\{ & f[u;\ell,r;A'] + f[w;\ell,r;A''] \\ & - c_{l(v)}\ell - c_{r(v)}r\colon\ 0 \leqslant A',\ 0 \leqslant A'', \\ & A' + A'' = A + a_{\ell(v)}\ell + a_{r(v)}r\}. \end{aligned}$$

In this formula, the value A' measures the total weight of the vertices in $G(u)$, and the value A'' measures the total weight of the vertices in $G(w)$. We impose $A' + A'' = A + a_{\ell(v)}\ell + a_{r(v)}r$, since in the total weight in $G(u)$ plus the total weight in $G(w)$, the weights of vertices $\ell(v)$ and $r(v)$ are counted twice. The terms $c_{\ell(v)}\ell$ and $c_{r(v)}r$ are subtracted, since they contribute to both terms $f[u;\ell,r;A']$ and $f[w;\ell,r;A'']$. Observe that if there is an edge between the two terminal vertices $\ell(v)$ and $r(v)$, then it either shows up only in $G(u)$ or only in $G(w)$. In either case, the term $c_{\ell(v),r(v)}\ell r$ is counted correctly in the formula.

In case v is an s vertex with left child u and right child w, we denote by $m(v)$ the right terminal of $G(u)$. Note that $m(v)$ simultaneously is the left terminal of $G(w)$. We set

$$\begin{aligned} f[v;\ell,r;A] := \max\{ & f[u;\ell,m;A'] + f[w;m,r;A''] \\ & - c_{m(v)}m\colon\ m \in \{0,1\},\ 0 \leqslant A',\ 0 \leqslant A'', \\ & A' + A'' + A + a_{m(v)}m\}. \end{aligned}$$

In this formula, the value A' measures the total weight of the vertices in $G(u)$, and the value A'' measures the total weight of the vertices in $G(w)$. We impose $A' + A'' = A + a_{m(v)}m$, since the total weight in $G(u)$ plus the total weight in $G(w)$ counts the weight of vertex $m(v)$ twice. The term $c_{m(v)}m$ has to be subtracted, since it contributes to $f[u;\ell,m;A']$ and to $f[w;m,r;A'']$.

In the very end, the global solution to the QKP instance can easily be computed from the values $f[root;\ell,r;A]$ where *root* denotes the root of the given binary decomposition tree. The optimal objective value equals

$$\max\{f[root;\ell,r;A]\colon\ \ell,r \in \{0,1\},\ A \leqslant b\}.$$

Now let us analyze the running time of this dynamic program. Since an ESP graph has at most $2n-1$ edges, the binary decomposition tree has at most $2n-1$ leaves and at most $2n-2$ interior vertices. Since the number of vertices v in the decomposition tree is $O(n)$, since $\ell, r \in \{0,1\}$, and since $0 \leqslant A \leqslant b$, there are $O(nb)$ values $f[v;\ell,r;A]$ that have to be computed. The computation of each such value according to the above formulas can be done in $O(A) = O(b)$ steps. Hence, the overall running time is $O(nb^2)$.

Theorem 3.2. *The special case of QKP where the underlying graph $G = (V,E)$ is ESP can be solved in pseudo-polynomial time $O(nb^2)$.*

3.2. The second DP algorithm

In this subsection, we design an algorithm that is pseudo-polynomial in the costs, and polynomial in the weights.

The approach is essentially dual to the above dynamic programming approach, with the roles of weights and costs exchanged. Hence, we will only briefly sketch this second approach and leave most of the details to the reader. Now consider an instance of QKP where the underlying graph $G = (V,E)$ is ESP. We define

$$U = \sum_{i \in V} |c_i| + \sum_{(i,j) \in E} |c_{ij}|. \tag{1}$$

Note that for any feasible solution $\mathbf{x}$ of QKP, the objective value satisfies $-U \leqslant F(\mathbf{x}) \leqslant U$. The starting point of our algorithm is again a binary decomposition tree of G. For a vertex v in this decomposition tree, we define $G(v)$, $\ell(v)$, $r(v)$, and QKP(v) exactly as above.

Definition 3.3. For a vertex v in some fixed binary decomposition tree, for $\ell, r \in \{0,1\}$, and for an integer C with $-U \leqslant C \leqslant U$, we denote by $g[v;\ell,r;C]$ the smallest possible weight for QKP(v) over all feasible solutions that satisfy the conditions

- $x_{\ell(v)} = \ell$ and $x_{r(v)} = r$, and
- $\sum_i c_i x_i + \sum_{(i,j)} c_{ij} x_i x_j = C$ where the summations are done over all the vertices i in $G(v)$, and over all the edges (i,j) in $G(v)$, respectively.

If there is no feasible solution of this form, then $g[v;\ell,r;C] = \infty$.

In the initialization phase of the dynamic program, we compute $g[v;\ell,r;C]$ for the leaves v of the decomposition tree:

$$g[v;\ell,r;c_{\ell(v)}\ell + c_{r(v)}r + c_{\ell(v),r(v)}\ell r] = a_{\ell(v)}\ell + a_{r(v)}r$$

and $g[v;\ell,r;C] = \infty$ whenever $C \neq c_{\ell(v)}\ell + c_{r(v)}r + c_{\ell(v),r(v)}\ell r$. If v is a p vertex with left child u and right child w, then

$$\begin{aligned} g[v;\ell,r;C] := \min\{&g[u;\ell,r;C'] + g[w;\ell,r;C''] \\ &- a_{\ell(v)}\ell - a_{r(v)}r : -U \leqslant C' \leqslant U,\ -U \leqslant C'' \leqslant U, \\ &C' + C'' = C + c_{\ell(v)}\ell + c_{r(v)}r\}. \end{aligned}$$

In case v is an s vertex with left child u and right child w, we denote by $m(v)$ the right terminal of $G(u)$. Then

$$\begin{aligned} g[v;\ell,r;C] := \min\{&g[u;\ell,m;C'] + g[w;m,r;C''] \\ &- a_{m(v)}m :\ m \in \{0,1\},\ -U \leqslant C' \leqslant U,\ -U \leqslant C'' \leqslant U, \\ &C' + C'' = C + c_{m(v)}m\}. \end{aligned}$$

In the end, the optimal objective value is the maximum C for which there exist $\ell, r \in \{0,1\}$ with $g[root;\ell,r;C] \leqslant b$. The overall running time of this dynamic program is $O(nU^2)$.

Theorem 3.4. *The special case of QKP where the underlying graph $G = (V,E)$ is ESP can be solved in pseudo-polynomial time $O(nU^2)$.*

4. A fully polynomial time approximation scheme

In this section, we design an FPTAS for the special case of QKP where all cost coefficients are non-negative and where the underlying graph $G = (V,E)$ is ESP. Our result crucially depends on the non-negativity of the cost coefficients. In fact without this non-negativity condition, the existence of an FPTAS would imply P = NP; see Theorem 5.2.

The FPTAS is based on the input rounding technique of Ibarra and Kim [12]. For an instance I of QKP with an underlying ESP graph $G = (V,E)$, we define the lower bound

$$L = \max\left\{\max_{i \in V} c_i,\ \max_{(i,j) \in E} c_i + c_j + c_{ij}\right\} \tag{2}$$

on the optimal objective value: Indeed, setting $x_i = 1$ and $x_k = 0$ for all $k \neq i$ yields a feasible solution $\mathbf{x}$ with objective value c_i; recall from the introduction that we assume $0 \leqslant a_i \leqslant b$ for all $i \in V$. And setting $x_i = x_j = 1$ and $x_k = 0$ for all $k \notin \{i,j\}$ yields for every edge $(i,j) \in E$ a feasible solution with objective value $c_i + c_j + c_{ij}$; recall from the introduction that we assume $a_i + a_j \leqslant b$ for all $(i,j) \in E$. If $L = 0$, then all c_i and all c_{ij} are 0 and the QKP is trivial to solve. Hence, we will from now on assume that $L \geqslant 1$.

Now let ε be the desired precision of approximation, and define a scaling parameter K by

$$K = \frac{\varepsilon L}{3n}.$$

Based on this parameter K, we define a rounded instance $I^{\#}$ for I with the same underlying ESP graph $G = (V,E)$, with the same weight coefficients a_i, the same weight bound b, and with the rounded cost coefficients

$$c_i^{\#} = \lfloor c_i/K \rfloor, \qquad c_{ij}^{\#} = \lfloor c_{ij}/K \rfloor$$

for $i \in V$ and $(i,j) \in E$. We compute the optimal solution $\mathbf{x}^{\#}$ for the rounded instance $I^{\#}$ according to Theorem 3.4. The output $\mathbf{x}^A$ of our approximation algorithm for instance I is either this feasible solution $\mathbf{x}^{\#}$, or a feasible solution that yields the objective value L, whichever has the largest objective value. This completes the description of our approximation algorithm.

Lemma 4.1. *Let $\mathbf{x}^*$ be the optimal solution for instance I, and let $\mathbf{x}^A$ be the solution found by our approximation algorithm. If $F(\mathbf{x})$ denotes the objective value of solution $\mathbf{x}$ for the original instance I, then*

$$F(\mathbf{x}^A) \geqslant (1 - \varepsilon) F(\mathbf{x}^*). \tag{3}$$

Proof. Denote by $F^{\#}(\mathbf{x})$ the objective value of solution $\mathbf{x}$ for the rounded instance $I^{\#}$. Since $c_i \geqslant Kc_i^{\#}$ and $c_{ij} \geqslant Kc_{ij}^{\#}$,

$$F(\mathbf{x}^{\#}) \geqslant KF^{\#}(\mathbf{x}^{\#}). \tag{4}$$

Since $\mathbf{x}^{\#}$ is the optimal solution for the rounded instance $I^{\#}$,

$$F^{\#}(\mathbf{x}^{\#}) \geqslant F^{\#}(\mathbf{x}^{*}). \tag{5}$$

Since $Kc_i^{\#} \geqslant c_i - K$ and $Kc_{ij}^{\#} \geqslant c_{ij} - K$, and since there are at most $3n$ terms in the objective function,

$$KF^{\#}(\mathbf{x}^{*}) \geqslant F(\mathbf{x}^{*}) - 3nK. \tag{6}$$

Putting (4)–(6) together yields

$$F(\mathbf{x}^{\#}) \geqslant F(\mathbf{x}^{*}) - 3nK = F(\mathbf{x}^{*}) - \varepsilon L. \tag{7}$$

The approximation $\mathbf{x}^{A}$ found by our approximation algorithm satisfies $F(\mathbf{x}^{A}) \geqslant F(\mathbf{x}^{\#})$ and $F(\mathbf{x}^{A}) \geqslant L$. By combining these bounds with (7) we get

$$F(\mathbf{x}^{A}) \geqslant F(\mathbf{x}^{*}) - \varepsilon F(\mathbf{x}^{A}),$$

which finally implies (3). □

Lemma 4.2. *The running time of the approximation algorithm is polynomial in n and* $1/\varepsilon$.

Proof. By Theorem 3.4, the optimal solution $\mathbf{x}^{\#}$ for instance $I^{\#}$ can be computed in $O(nU^2)$ time, where the value U is defined as in Eq. (1) as

$$\begin{aligned} U &= \sum_{i\in V} |c_i^{\#}| + \sum_{(i,j)\in E} |c_{ij}^{\#}| \\ &= \sum_{i\in V} \lfloor c_i/K \rfloor + \sum_{(i,j)\in E} \lfloor c_{ij}/K \rfloor \\ &\leqslant \frac{1}{K}\left(\sum_{i\in V} c_i + \sum_{(i,j)\in E} c_{ij}\right) \\ &\leqslant \frac{1}{K}(|V| + |E|)L \leqslant \frac{3nL}{K} = \frac{9n^2}{\varepsilon}. \end{aligned}$$

Here we have first used the definition of $c_i^{\#}$ and $c_{ij}^{\#}$, then the fact that in an ESP graph $|E| \leqslant 2|V| - 1$, and finally the definition of K. We conclude that the running time $O(nU^2)$ is $O(n^5/\varepsilon^2)$ and hence polynomially bounded in n and $1/\varepsilon$. □

Lemmas 4.1 and 4.2 together show that our approximation algorithm indeed is an FPTAS.

Theorem 4.3. *The special case of QKP where all cost coefficients are non-negative and where the underlying graph* $G = (V, E)$ *is ESP possesses a fully polynomial time approximation scheme.*

It is instructive to check where we have actually exploited the non-negativity of the cost coefficients: It can be verified that all our arguments go through as long as the value L defined in (2) is strictly positive. Moreover, the case $L = 0$ is trivial to solve, and the case of negative L cannot occur for non-negative cost coefficients. For arbitrary cost coefficients, however, even the case $L = 0$ is in-approximable unless P = NP; see Theorem 5.2.

5. Negative results

In this section, we prove two negative results on the complexity and the approximability of special cases of the QKP: the QKP with vertex series–parallel graphs is strongly NP-hard, and the QKP with negative cost coefficients is in-approximable in a very strong way unless P = NP.

Theorem 5.1. *The special case of QKP where the underlying graph* $G = (V, E)$ *is vertex series–parallel is NP-hard in the strong sense.*

Proof. The proof is done by a reduction from BALANCED COMPLETE BIPARTITE SUBGRAPH (BCBS, for short); this is problem [GT24] in Garey and Johnson [9]. An instance of BCBS consists of a bipartite graph $H = (V_1 \cup V_2, E_H)$ together with a positive integer k. The question is whether there exist subsets $W_1 \subseteq V_1$ and $W_2 \subseteq V_2$ with $|W_1| = |W_2| = k$ that span a complete bipartite subgraph of H. BCBS is NP-complete.

We construct the following instance of QKP. Without loss of generality let $V_1 \cup V_2 = \{1, \ldots, n\}$. For $i = 1, \ldots, n$ we set $c_i = 0$ and $a_i = 1$. Moreover, we set $b = 2k$ and $E = V_1 \times V_2$. For $(i, j) \in E$, we set $c_{ij} = 2$ if $(i, j) \in E_H$ and $c_{ij} = 1$ if $(i, j) \notin E_H$. Note that for the constructed QKP instance the underlying graph is a complete bipartite graph, and hence a vertex series–parallel graph. Note furthermore that all the cost and weight coefficients are polynomially bounded in the input; hence, this reduction will indeed establish strong NP-hardness. We claim that the BCBS instance

has answer YES, if and only if this QKP instance has a feasible solution with objective value at least $2k^2$.

Proof of (if): Let $\mathbf{x}$ be a feasible solution for QKP with objective value at least $2k^2$. Define $W_1 \subseteq V_1$ and $W_2 \subseteq V_2$ to contain the vertices i with $x_i = 1$. By the weight constraint, $|W_1| + |W_2| \leqslant b = 2k$. Let ℓ denote the number of edges $(i,j) \in W_1 \times W_2$ that are not in E_H. Then the objective value for $\mathbf{x}$ is

$$2k^2 = 2|W_1|\,|W_2| - \ell \leqslant 2|W_1|(2k - |W_1|) - \ell \leqslant 2k^2 - \ell.$$

This yields $\ell = 0$ and $|W_1| = |W_2| = k$.

Proof of (only if): Let W_1 and W_2 be a certificate for BCBS. Then setting $x_i = 1$ if and only if $i \in W_1 \cup W_2$ yields a feasible solution $\mathbf{x}$ for QKP with objective value at least $2k^2$. The proof is complete. □

Theorem 5.2. *Unless $P = NP$, the special case of QKP where the underlying graph $G = (V, E)$ is a tree and where the cost coefficients c_i and c_{ij} may be negative does not have any polynomial time approximation algorithm with finite worst case guarantee.*

Proof. The proof is done by a reduction from the NP-complete SUBSET SUM problem; this is problem [SP13] in Garey and Johnson [9]. An instance consists of positive integers $q_1, \ldots, q_n$ and Q. The question is whether there exists a subset $I \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in I} q_i = Q$.

We construct the following instance of QKP: the underlying graph is a star with n leaves. There is a central vertex 0, and there are n edges $(0, i)$ to the other vertices $i = 1, \ldots, n$. The central vertex 0 has cost $c_0 = -Q + 1$, and all other vertices i $(i = 1, \ldots, n)$ have cost $c_i = 0$. The cost of the edge $(0, i)$ equals $c_{0i} = q_i$. Moreover, $b = Q$ and the weight coefficients are $a_0 = 0$ and $a_i = q_i$ $(i = 1, \ldots, n)$.

Consider a feasible solution $\mathbf{x}$ for this QKP instance. If $x_0 = 0$, then $F(\mathbf{x}) = 0$. If $x_0 = 1$, then $F(\mathbf{x}) = -Q + 1 + q(\mathbf{x})$ where $q(\mathbf{x}) = \sum_i q_i x_i$. Because of the weight constraint $q(\mathbf{x}) \leqslant b = Q$, and hence $F(\mathbf{x}) = 1$ if and only if the SUBSET SUM instance has answer YES. To summarize, this QKP instance has optimal objective value 1 if the SUBSET SUM instance has answer YES, and otherwise its optimal objective value is 0. Any polynomial time approximation algorithm with finite worst case guarantee could be used to distinguish between the objective values 0 and 1 in polynomial time. This would imply a polynomial time solution for SUBSET SUM. □

6. Conclusion

In this paper, we have derived two pseudo-polynomial time algorithms for the special case of the quadratic 0–1 knapsack problem where the underlying graph is edge series–parallel. One of these algorithms could be turned into a fully polynomial time approximation scheme for the case where additionally all the cost coefficients are non-negative. It does not help a lot, however, if the underlying graph is vertex series–parallel: We proved that this special case of the QKP is strongly NP-complete.

The time complexity of our first pseudo-polynomial time algorithm is $O(nb^2)$. For paths and for cycles, it is quite easy to get algorithms with a better time complexity $O(nb)$. So the question arises, whether there exists an $O(nb)$ time algorithm for general edge series–parallel graphs.

Another contribution of this paper is that, unlike previous work, the structure of the graph G naturally defined by the objective function plays a major role in the solution of the problem. It is hoped that, in future work, the structure of this graph can be further exploited to derive algorithms for QKP.

References

[1] A. Billionnet, Maximizing a quadratic tree-structured pseudo-boolean function with a cardinality constraint, Manuscript.

[2] A. Billionnet, F. Calmels, Linear programming for the 0–1 quadratic knapsack problem, Eur. J. Oper. Res. 92 (1996) 310–325.

[3] A. Billionnet, A. Faye, E. Soutif, A new upper-bound for the 0–1 quadratic knapsack problem, Eur. J. Oper. Res. 112 (1999) 664–672.

[4] A. Caprara, D. Pisinger, P. Toth, Exact solution of the quadratic knapsack problem, INFORMS J. Comput. 11 (1999) 125–137.

[5] P. Chaillou, P. Hansen, Y. Mahieu, Best Network Flow Bounds for the Quadratic Knapsack Problem, Lecture Notes in Mathematics, Vol. 1403, Springer, New York, 1986, pp. 226–235.

[6] R.J. Duffin, Topology of series-parallel networks, J. Math. Anal. Appl. 10 (1965) 303–318.

[7] G. Gallo, P.L. Hammer, B. Simeone, Quadratic knapsack problems, Math. Prog. 12 (1980) 132–149.

[8] G. Gallo, S. Simeone, On the supermodular knapsack problem, Math. Prog. 45 (1988) 295–309.
[9] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, 1979.
[10] P.L. Hammer, D.J. Rader, Efficient methods for solving quadratic 0–1 knapsack problems, INFOR 35 (1997) 170–182.
[11] C. Helmberg, F. Rendl, R. Weismantel, Quadratic knapsack relaxations using cutting planes and semidefinite programming, Lecture Notes in Computer Science, Vol. 1084, Springer, Berlin, 1996, pp. 175–189.
[12] O. Ibarra, C.E. Kim, Fast approximation algorithms for the knapsack and sum of subset problems, J. ACM 22 (1975) 463–468.
[13] E.L. Johnson, A. Mehrotra, G.L. Nemhauser, Min-cut clustering, Math. Prog. 62 (1993) 133–151.
[14] E.L. Lawler, Fast approximation schemes for knapsack problems, Math. Oper. Res. 4 (1979) 339–356.
[15] P. Michelon, L. Veilleux, Lagrangean methods for the 0–1 quadratic knapsack problem, European J. Oper. Res. 92 (1996) 326–341.
[16] D.J. Rader, Valid inequalities and facets of the quadratic 0–1 knapsack polytope, RUTCOR Research Report 16-97, RUTCOR, Rutgers University, 1997.
[17] K. Takamizawa, T. Nishizeki, N. Saito, Linear-time computability of combinatorial problems on series-parallel graphs, J. ACM 29 (1982) 623–641.
[18] J. Valdes, R.E. Tarjan, E.L. Lawler, The recognition of series-parallel digraphs, SIAM J. Comput. 11 (1982) 298–313.
[19] C. Witzgall, Mathematical methods of site selection for electronic message systems (EMS), NBS Internal Report, 1975.