Approximation schemes for a class of subset selection problems

Kirk Pruhs[1]* and Gerhard J. Woeginger[2]

[1] Department of Computer Science, University of Pittsburgh, USA
kirk@cs.pitt.edu
WWW home page: **http://www.cs.pitt.edu/~kirk**
[2] Department of Mathematics, University of Twente, The Netherlands
g.j.woeginger@math.utwente.nl
WWW home page: **http://wwwhome.cs.utwente.nl/~woegingergj**

Abstract. In paper we develop an easily applicable algorithmic technique/tool for developing approximation schemes for certain types of combinatorial optimization problems. Special cases that are covered by our result show up in many places in the literature. For every such special case, a particular rounding trick has been implemented in a slightly different way, with slightly different arguments, and with slightly different worst case estimations. Usually, the rounding procedure depended on certain upper or lower bounds on the optimal objective value that have to be justified in a separate argument. Our easily applied result unifies many of these results, and sometimes it even leads to a simpler proof.
We demonstrate how our result can be easily applied to a broad family of combinatorial optimization problems. As a special case, we derive the existence of an FPTAS for the scheduling problem of minimizing the weighted number of late jobs under release dates and preemption on a single machine. The approximability status of this problem has been open for some time.

1 Introduction

One of the commonly stated goals of algorithmic research is the development of a modestly-sized toolkit of widely applicable algorithmic techniques. The vision is that future researchers, particularly those without specialized training in algorithmics, could use these tools to quickly develop/analyze algorithms for new problems. In this paper, we develop an easily and widely applicable algorithmic technique/tool for developing approximation schemes for certain types of combinatorial optimization problems. This tool should save algorithmic researchers time, and is simple enough to be used by researchers without specialized algorithmics training.

Over the years, there have evolved a number of standard approaches for designing approximation schemes; see for instance Horowitz & Sahni [8, 9], Ibarra

* Supported in part by NSF grant CCR-0098752, NSF grant ANI-0123705, and NSF grant ANI-0325353.

& Kim [10], Sahni [18], and Woeginger [20]. (A review of basic definitions related to approximation schemes can be found in section 2.) We will investigate one of these standard approaches and demonstrate that it applies to a broad family of combinatorial optimization problems. The standard approach under investigation is the technique of *rounding the input*; this technique goes back to the 1970s and possibly has first been used in the paper by Horowitz & Sahni [8]. The family of combinatorial optimization problems under investigation is defined as follows.

Definition 1. *(Subset selection problems)*
A subset selection problem $\mathcal{P}$ is a combinatorial optimization problem whose instances $I = (X, w, S)$ consist of

- *a ground set X with $|X| = n$ elements;*
- *a positive integer weight $w(x)$ for every $x \in X$;*
- *a structure S that is described by $\ell(S)$ bits;*

The structure S specifies for every subset $Y \subseteq X$ whether Y is feasible *or* infeasible*; this can be done within a time complexity polynomially bounded in n and $\ell(S)$.*

If $\mathcal{P}$ is a minimization *problem, then the goal is to find a feasible subset $Y \subseteq X$ that minimizes $w(Y) \doteq \sum_{y \in Y} w(y)$, and if $\mathcal{P}$ is a* maximization *problem, then the goal is to find a feasible subset $Y \subseteq X$ that maximizes $w(Y)$.* □

The class of subset selection problems described in Definition 1 is very general, and it contains many problems with very bad approximability behavior. For instance, the *weighted independent set* problem ("Given a graph with vertex weights, find the maximum weight subset of pairwise non-adjacent vertices") belongs to this class. It is known that weighted independent set does not possess *any* ρ-approximation algorithm with a fixed $\rho \geq 1$, unless P=NP (Håstad [7]). If we additionally impose condition (C) as in the following theorem, then the approximability behavior of subset selection problems improves considerably.

Theorem 1. *Let $\mathcal{P}$ be a subset selection problem with instances $I = (X, w, S)$ that satisfies the following condition:*

(C) There exists an algorithm that solves $\mathcal{P}$ to optimality whose running time is polynomially bounded in n, in $W := \sum_{x \in X} w(x)$, and in $\ell(S)$.

Then problem $\mathcal{P}$ has an FPTAS.

Theorem 1 is proved in Section 3. The proof is quite straightforward, and it mainly uses the folklore rounding tricks from the literature. The main contribution of this paper is to identify the neat and simple condition (C) that automatically implies the existence of an FPTAS. Special cases that are covered by Theorem 1 show up at many places in the literature. For every such special case, the rounding trick has been implemented in a slightly different way, with slightly different arguments, and with slightly different worst case estimations.

Usually, the rounding procedure depends on certain upper or lower bounds on the optimal objective value that have to be justified in a separate argument. Theorem 1 unifies many of these results, and sometimes it even leads to simpler proofs.

Sections 4 and 5 contain a number of optimization problems (from scheduling theory and from graph theory) that fit into the framework of Definition 1. These examples illustrate the wide applicability and ease of use of our result. As one special case, we prove in Theorem 2 that the scheduling problem $1 \,|\, pmtn, r_j \,|\, \sum w_j U_j$ (the problem of minimizing the weighted number of late jobs under release dates and preemption on a single machine) has a PTAS. The approximability status of this problem has been open for some time, and the question was considered to be difficult. In particular, the problem does not fit into the framework for FPTAS's established by Woeginger [20].

2 Basic Definitions

An algorithm that returns near-optimal solutions is called an *approximation algorithm*; if it does this in polynomial time, then it is called a *polynomial time* approximation algorithm. An approximation algorithm is called a *ρ-approximation algorithm*, if it always returns a near-optimal solution with cost at most a factor ρ above the optimal cost (for minimization problems) respectively at most a factor ρ below the optimal cost (for maximization problems). The value $\rho \geq 1$ is called the *worst-case performance guarantee* of this algorithm. A family of $(1+\varepsilon)$-approximation algorithms over all real $\varepsilon > 0$ with polynomial running times is called a *polynomial time approximation scheme* or PTAS, for short. If the time complexity of a PTAS is also polynomially bounded in $1/\varepsilon$, then it is called a *fully* polynomial time approximation scheme or FPTAS, for short. With respect to relative performance guarantees, an FPTAS is essentially the strongest possible polynomial time approximation result that we can derive for an NP-hard problem (unless P=NP holds).

3 Proof of the main result

In this section we will prove the main result of the paper. The proof method is essentially due to Horowitz & Sahni [8]. The arguments for minimization problems and for maximization problems are slightly different. We start with the discussion of minimization problems.

Let $\varepsilon > 0$ be a small real number. Let $I = (X, w, S)$ be some instance of a minimization problem $\mathcal{P}$ that belongs to the class of subset selection problems as defined in Definition 1. Let $x_1, \ldots, x_n$ be an enumeration of the elements of the ground set X such that

$$w(x_1) \;\leq\; w(x_2) \;\leq\; \cdots \;\leq\; w(x_n). \tag{1}$$

For $k = 1, \ldots, n$, we define a so-called scaling parameter

$$Z^{(k)} \ \doteq \ \varepsilon \cdot \frac{1}{n} \cdot w(x_k). \tag{2}$$

We introduce a number of new instances $I^{(1)}, \ldots, I^{(n)}$ of problem $\mathcal{P}$. Every new instance $I^{(k)}$ has the same structure S and the same ground set X as I, but it has a different set of weights $w^{(k)}$. As a consequence, all instances $I^{(k)}$ have the same feasible solutions as the original instance I. The weights $w^{(k)}$ are defined as follows:

- For $i = 1, \ldots, k$, we set $w^{(k)}(x_i) = \lceil w(x_i)/Z^{(k)} \rceil$.
- For $i = k+1, \ldots, n$, we set $w^{(k)}(x_i) = n\lceil n/\varepsilon \rceil$.

The definition of the parameter $Z^{(k)}$ in (2) yields that $w^{(k)}(x_i) \leq \lceil n/\varepsilon \rceil$ for $1 \leq i \leq k$. Therefore, the overall weight $W^{(k)}$ of all elements in instance $I^{(k)}$ can be bounded as

$$W^{(k)} \ \leq \ \sum_{i=1}^{k} \lceil n/\varepsilon \rceil + \sum_{i=k+1}^{n} n\lceil n/\varepsilon \rceil \ \leq \ n^2 \lceil n/\varepsilon \rceil. \tag{3}$$

Hence, $W^{(k)}$ is polynomially bounded in n and in $1/\varepsilon$. If we feed instance $I^{(k)}$ to the exact algorithm in condition (C) in Theorem 1, then the running time is polynomially bounded in n, in $\ell(S)$, and in $1/\varepsilon$. That's precisely the type of time complexity that we need for an FPTAS. Hence we get Lemma 1.

Lemma 1. *Every instance $I^{(k)}$ can be solved to optimality within a time complexity polynomially bounded in n, $\ell(S)$, and $1/\varepsilon$.* □

Next, let Y^* denote the optimal solution for the original instance I, and let Opt denote its optimal objective value $w(Y^*)$. Let $Y^{(k)} \subseteq X$ denote the optimal solution for instance $I^{(k)}$ for $k = 1, \ldots, n$. Let j denote the maximal index with $x_j \in Y^*$. Obviously,

$$\text{Opt} \ = \ w(Y^*) \ \geq \ w(x_j). \tag{4}$$

Furthermore, we claim that

$$Y^{(j)} \ \subseteq \ \{x_1, x_2, \ldots, x_j\}. \tag{5}$$

This statement is vacuously true for $j = n$. For $j \leq n-1$, we use that Y^* is *some* feasible solution for instance $I^{(j)}$, whereas $Y^{(j)}$ is the *optimal* feasible solution for instance $I^{(j)}$. Since $|Y^*| \leq j \leq n-1$, this yields

$$w^{(j)}(Y^{(j)}) \ \leq \ w^{(j)}(Y^*) \ \leq \ |Y^*| \cdot w^{(j)}(x_j) \ \leq \ (n-1) \cdot \lceil n/\varepsilon \rceil. \tag{6}$$

By Inequality (6), the set $Y^{(j)}$ can not contain any of the expensive elements $x_{j+1}, \ldots, x_n$ that all have weight $n\lceil n/\varepsilon \rceil$. This proves (5).

We now analyze the quality of the feasible solution $Y^{(j)}$ for the original instance I. In the following chain of inequalities, the first inequality holds since

(5) implies $w(y) \le Z^{(j)} \cdot w^{(j)}(y)$ for all $y \in Y^{(j)}$. The second inequality holds, since $Y^{(j)}$ is the optimal solution for weights $w^{(j)}(\cdot)$. The equation in the third line follows from (5). The inequality in the fourth line follows from $\lceil\alpha\rceil \le \alpha + 1$. The inequality in the sixth line follows from $|Y^*| \le n$ and from (2). The final inequality follows from (4).

$$\begin{aligned}
\sum\{w(y) \mid y \in Y^{(j)}\} &\le Z^{(j)} \cdot \sum \{ w^{(j)}(y) \mid y \in Y^{(j)} \} \\
&\le Z^{(j)} \cdot \sum \{ w^{(j)}(y) \mid y \in Y^* \} \\
&= Z^{(j)} \cdot \sum \{ \lceil w(y)/Z^{(j)} \rceil \mid y \in Y^* \} \\
&\le Z^{(j)} \cdot \sum \{ w(y)/Z^{(j)} + 1 \mid y \in Y^* \} \\
&= \sum \{ w(y) \mid y \in Y^* \} + |Y^*| \cdot Z^{(j)} \\
&\le \textsc{Opt} + n \cdot \varepsilon \cdot \frac{1}{n} \cdot w(x_j) \\
&\le (1+\varepsilon) \cdot \textsc{Opt}.
\end{aligned}$$

With this, it is clear how to get the FPTAS: We compute the optimal solutions $Y^{(k)} \subseteq X$ for the instances $I^{(k)}$ with $k = 1, \ldots, n$. By Lemma 1, this can be done with time complexity polynomially bounded in the length of the encoding of I and in $1/\varepsilon$. Then we compute the costs of $Y^{(k)}$ with respect to instance I, and we determine the best solution. By the above chain of inequalities, this best solution has objective value at most $(1+\varepsilon)\textsc{Opt}$. This completes the proof of Theorem 1 for the case where $\mathcal{P}$ is a minimization problem.

Now let us discuss the case where $\mathcal{P}$ is a maximization problem. Consider an instance $I = (X, w, S)$ of $\mathcal{P}$, and enumerate the elements of the ground set X as in (1). In a preprocessing phase, we determine for every element x_k $(k = 1, \ldots, n)$ whether there exists a feasible solution that contains x_k. This can be done as follows: We create a new instance $I^{(k)}$ from I by setting the weight of element x_k to n and by setting the weights of the remaining $n-1$ elements to 1. Clearly, x_k shows up in some feasible solution if and only if the optimal objective value of $I^{(k)}$ is greater or equal to n. Since the overall weight in instance $I^{(k)}$ is $2n-1$, the algorithm from condition (C) can be used to solve it in time polynomially bounded in n and $\ell(S)$.

The main part of our algorithm is built around the maximal index j for which element x_j occurs in some feasible solution. This implies

$$\textsc{Opt} \ \ge \ w(x_j). \tag{7}$$

We introduce a scaling parameter $Z^{\#} \doteq \varepsilon \cdot \frac{1}{n} \cdot w(x_j)$. We define a new instance $I^{\#}$ from I that has new weights $w^{\#}$. For $i = 1, \ldots, j$ we set $w^{\#}(x_i) = \lfloor w(x_i)/Z^{\#} \rfloor$, and for $i = j+1, \ldots, n$ we set $w^{\#}(x_i) = 1$. Similarly as in the minimization case, instance $I^{\#}$ can be solved to optimality within a time complexity polynomially bounded in n, $\ell(S)$, and $1/\varepsilon$.

The optimal solution $Y^{\#}$ of $I^{\#}$ satisfies the following inequalities. These inequalities run in parallel to the inequalities for the minimization case. In the third line, we use (7) to bound $w(x_j)$.

$$\begin{aligned}
\sum\{w(y) \mid y \in Y^{\#}\} &\geq Z^{\#} \cdot \sum \{ \, w^{\#}(y) \mid y \in Y^{\#} \, \} \\
&\geq Z^{\#} \cdot \sum \{ \, w^{\#}(y) \mid y \in Y^{*} \, \} \\
&= Z^{\#} \cdot \sum \{ \, \lfloor w(y)/Z^{\#} \rfloor \mid y \in Y^{*} \, \} \\
&\geq Z^{\#} \cdot \sum \{ \, w(y)/Z^{\#} \; - 1 \mid y \in Y^{*} \, \} \\
&= \sum \{ \, w(y) \mid y \in Y^{*} \, \} \; - \; |Y^{*}| \cdot Z^{\#} \\
&\geq \textsc{Opt} \; - \; n \cdot \varepsilon \cdot \frac{1}{n} \cdot w(x_j) \\
&\geq (1-\varepsilon) \cdot \textsc{Opt}.
\end{aligned}$$

Hence, also maximization problems have an FPTAS. The proof of Theorem 1 is complete.

4 Example: Scheduling to minimize the weighted number of late jobs

In this section, we will use the standard three-field scheduling notation (see e.g. Graham, Lawler, Lenstra & Rinnooy Kan [5] and Lawler, Lenstra, Rinnooy Kan & Shmoys [14]).

In the scheduling problem $1 \, | \, | \sum w_j U_j$, the input consists of n jobs J_j with positive integer processing times p_j, weights w_j, and due dates d_j $(j = 1, \ldots, n)$. All jobs are available for processing at time 0. In some schedule a job is *on-time* if its processing is completed by its deadline, and otherwise it is *late*. The goal is to schedule the jobs without interruption on a single machine such that the total weight of the late jobs is minimized. The problem $1 \, | \, | \sum w_j U_j$ is known to be NP-hard in the ordinary sense (Karp [11]).

Problem $1 \, | \, | \sum w_j U_j$ belongs to the class of subset selection problems as described in Definition 1. The ground set X consists of the n jobs with weights w_k and total weight $W = \sum_{i=1}^{n} w_k$. The structure S consists of the processing times p_j and the due dates d_j $(j = 1, \ldots, n)$. A subset Y of the jobs is feasible, if the remaining jobs in $X - Y$ can all be scheduled on-time on a single machine; clearly, this information is specified by the structure S. Lawler & Moore [15] give a dynamic programming formulation that solves $1 \, | \, | \sum w_j U_j$ in $O(nW)$ time.

Then our main result in Theorem 1 implies the following well-known result of Gens & Levner [4].

Corollary 1. *(Gens & Levner [4], 1981)*
There exists an FPTAS for minimizing the weighted number of late jobs in the scheduling problem $1\,|\,|\sum w_j U_j$. □

A closely related problem is to maximize the total weight of the *on-time* jobs. Clearly, the algorithm of Lawler & Moore [15] also solves this maximization problem in $O(nW)$ time. We get the following result.

Corollary 2. *(Sahni [18], 1976)*
There exists an FPTAS for maximizing the weighted number of on-time jobs in the scheduling problem $1\,|\,|\sum w_j U_j$. □

In the *0/1-knapsack problem*, the input consists of n pairs of positive integers (w_k, b_k) and a positive integer b: The weight w_k denotes the profit of the kth item, and b_k denotes the space occupied by this item. The goal is to select a subset Y that has the maximum profit subject to the condition that it does not occupy more than b space. The *0/1-knapsack problem* is NP-hard (Karp [11]), and it can be solved in $O(nW)$ time (see for instance Bellman & Dreyfus [1] or Martello & Toth [17]).

It is easy to see that the 0/1-knapsack problem belongs to the class of subset selection problems of Definition 1. In fact, it is a special case of the maximization version of the scheduling problem $1\,|\,|\sum w_j U_j$ as described above: Essentially, the kth item corresponds to a job with processing time b_k, weight w_k, and (universal) due date b.

Corollary 3. *(Ibarra & Kim [10], 1975)*
The 0/1-knapsack problem possesses an FPTAS. □

Another closely related problem is $1\,|\,pmtn, r_j\,|\sum w_j U_j$: There are n jobs J_j $(j = 1, \ldots, n)$ with processing times p_j, weights w_j, due dates d_j, and release dates r_j. In this variant, job J_j cannot be started before its release date r_j, but it may be preempted. Lawler [13] designs a (very complicated) dynamic program that solves $1\,|\,pmtn, r_j\,|\sum w_j U_j$ in $O(n^3 W^2)$ time. We get the following (new) result.

Theorem 2. *There exists an FPTAS for minimizing the weighted number of late jobs in the scheduling problem* $1\,|\,pmtn, r_j\,|\sum w_j U_j$.

5 Example: The restricted shortest path problem

An instance of the restricted shortest path problem (RSP, for short) consists of a directed graph $G = (V, A)$ and an integer bound T. Every arc $a \in A$ has a positive integer cost w_a and a positive integer transition time t_a. For a directed path Y in G, the cost $w(Y)$ and the transition time $t(Y)$ are defined as the sum of the costs and transition times, respectively, of the edges in the path Y. The goal is to find a path Y with $t(Y) \leq T$ from a specified source vertex to a specified target vertex, that minimizes the cost. RSP is NP-complete in the ordinary sense (Garey & Johnson [3]). Furthermore, RSP is solvable in $O(|A| \cdot W)$ time by dynamic programming; see for instance Warburton [19] or Hassin [6]. One way of doing this is to compute for every vertex $v \in V$ and for every cost $c \in \{0, \ldots, W\}$, the smallest possible transition time of a path from the source vertex to v with cost c.

Problem RSP belongs to the class of subset selection problems as described in Definition 1. The ground set X consists of the arcs $a \in A$ with costs w_a. The structure S consists of the graph G, of the transition times t_j, of the bound T, and of the source and sink vertices. A subset Y of the arcs is feasible, if it forms a path from source to sink with transition time $t(Y) \leq T$. Obviously, this feasibility information is encoded by the structure S. We get the following result.

Corollary 4. *(Hassin [6], 1992)*
The restricted shortest path problem RSP possesses an FPTAS. □

The result in Corollary 4 has been established in 1987 by Warburton [19] for acyclic directed graphs and then in 1992 by Hassin [6] for arbitrary directed graphs. Lorenz & Raz [16] and Ergun, Sinha & Zhang [2] improve the time complexities of these approximation schemes.

References

1. R.E. Bellman and S.E. Dreyfus (1962). *Applied Dynamic Programming.* Princeton University Press.
2. F. Ergun, R. Sinha, and L. Zhang (2002). An improved FPTAS for restricted shortest path. *Information Processing Letters 83*, 287–291.
3. M.R. Garey and D.S. Johnson (1979). *Computers and Intractability.* W.H. Freeman and Co., New York.
4. G.V. Gens and E.V. Levner (1981). Fast approximation algorithms for job sequencing with deadlines. *Discrete Applied Mathematics 3*, 313–318.
5. R.L. Graham, E.L. Lawler, J.K. Lenstra, and A.H.G. Rinnooy Kan (1979). Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics 5*, 287–326.
6. R. Hassin (1992). Approximation schemes for the restricted shortest path problem. *Mathematics of Operations Research 17*, 36–42.
7. J. Håstad (1999). Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica 182*, 105–142.
8. E. Horowitz and S. Sahni (1974). Computing partitions with applications to the knapsack problem. *Journal of the ACM 21*, 277–292.
9. E. Horowitz and S. Sahni (1976). Exact and approximate algorithms for scheduling nonidentical processors. *Journal of the ACM 23*, 317–327.
10. O. Ibarra and C.E. Kim (1975). Fast approximation algorithms for the knapsack and sum of subset problems. *Journal of the ACM 22*, 463–468.
11. R.M. Karp (1972). Reducibility among combinatorial problems. In: R.E. Miller and J.W. Thatcher, editors, *Complexity of Computer Computations*, Plenum Press, New York, 85–104.

12. E.L. LAWLER (1979). Fast approximation schemes for knapsack problems. *Mathematics of Operations Research 4*, 339–356.
13. E.L. LAWLER (1990). A dynamic programming algorithm for preemptive scheduling of a single machine to minimize the number of late jobs. *Annals of Operations Research 26*, 125–133.
14. E.L. LAWLER, J.K. LENSTRA, A.H.G. RINNOOY KAN, AND D.B. SHMOYS (1993). Sequencing and scheduling: Algorithms and complexity. In: S.C. Graves, A.H.G. Rinnooy Kan, and P.H. Zipkin (eds.) *Logistics of Production and Inventory*, Handbooks in Operations Research and Management Science 4, North-Holland, Amsterdam, 445–522.
15. E.L. LAWLER AND J.M. MOORE (1969). A functional equation and its application to resource allocation and sequencing problems. *Management Science 16*, 77–84.
16. D.H. LORENZ AND D. RAZ (2001). A simple efficient approximation scheme for the restricted shortest path problem. *Operations Research Letters 28*, 213–219.
17. S. MARTELLO AND P. TOTH [1990]. Knapsack problems: Algorithms and computer implementations. John Wiley & Sons, England.
18. S. SAHNI (1976). Algorithms for scheduling independent tasks. *Journal of the ACM 23*, 116–127.
19. A. WARBURTON (1987). Approximation of pareto optima in multiple-objective shortest path problems. *Operations Research 35*, 70–79.
20. G.J. WOEGINGER (2000). When does a dynamic programming formulation guarantee the existence of a fully polynomial time approximation scheme (FPTAS)? *INFORMS Journal on Computing 12*, 57–75.