J Optim Theory Appl (2012) 152:378–393
DOI 10.1007/s10957-011-9902-7

Minimum and Worst-Case Performance Ratios of Rollout Algorithms

Luca Bertazzi

Received: 21 February 2011 / Accepted: 11 July 2011 / Published online: 9 August 2011

Abstract Rollout algorithms are heuristic algorithms that can be applied to solve deterministic and stochastic dynamic programming problems. The basic idea is to use the cost obtained by applying a well known heuristic, called the base policy, to approximate the value of the optimal cost-to-go. We develop a theoretical approach to prove, for the 0-1 knapsack problem, that the minimum performance ratio of the rollout algorithms tends to be significantly greater when the performance ratio of the corresponding base policy is poor and that the worst-case performance ratio is significantly better than the one of the corresponding base policies.

Keywords Rollout algorithms · Minimum performance ratio · Worst-case performance ratio

1 Introduction

Rollout algorithms are a class of heuristic algorithms more and more frequently applied to solve deterministic and stochastic dynamic programming problems. The basic idea is to use, in a one-step lookahead policy, the cost of a well known heuristic, called base policy, to approximate the value of the optimal cost-to-go.

These algorithms have been originally proposed in the context of Neuro-Dynamic Programming/Reinforcement Learning (see [1] and [2]). They have been applied to stochastic scheduling by [3], to vehicle routing problems with stochastic demand by

Communicated by Rainer Burkard.

The author wishes to thank the Associate Editor and two anonymous Referees for their useful comments and suggestions.

L. Bertazzi (✉)
Department of Quantitative Methods, University of Brescia, Contrada Santa Chiara 50, 25122 Brescia, Italy
e-mail: bertazzi@eco.unibs.it

[4–6] and [7]. Computational papers related to rollout algorithms can be found in [8] and [9]. They have been proposed for the solution of combinatorial problems by [1, 10] and [11]. They have been applied to multi–dimensional knapsack problems by [12]. Based on the breakthrough problem, [13] provides interesting insights into the nature of cost improvement of rollout algorithms.

These algorithms are very appealing from the practical point of view, as they are easy to be implemented and guarantee a not worse, and usually much better, performance than the corresponding base policies (see [13]). In all previous papers, the performance of the rollout algorithms has been evaluated on the basis of computational experiments. The computational results typically show that these algorithms are very effective on average with respect to the corresponding base policy. However, no theoretical results are available to show the performance of rollout algorithms, given the performance of the corresponding base policy.

Theoretical results are based on the concept of performance ratio and worst-case performance ratio. Consider a maximization problem. The performance ratio of a given algorithm in a given instance is the ratio between the profit obtained by the algorithm and the optimal profit, while the worst-case performance ratio of the algorithm is the infimum of this ratio over all instances.

Two questions are of particular interest when the performance of a rollout algorithm is studied from the theoretical point of view. The first is the following. Suppose to know the performance ratio of the base policy in a given instance. The question is: Is it possible to prove a minimum performance ratio of the corresponding rollout algorithm in the same instance? The answer to this question is very important, even from a practical point of view, as it allows one to estimate the minimum guaranteed performance of the rollout algorithm. The second question concerns the worst-case performance of the rollout algorithm. Suppose to know the worst-case performance of the base policy. The question is: Which is the worst-case performance of the corresponding rollout algorithm? The answer to this question is important, as it allows us to evaluate the improvement of the rollout algorithm with respect to the corresponding base policy, in the worst case. Given that the computational times of a rollout algorithm are greater than the ones of the corresponding base policy, the answer to this question allows us to evaluate ‘costs and benefits’ of the application of the rollout algorithm with respect to the corresponding base policy.

Our aim is to give an answer to these two questions for an ‘easy’ classical combinatorial problem: The 0-1 Knapsack Problem. Therefore, our aim is not to propose a new heuristic algorithm for the solution of this problem, but to improve our understanding about why and when rollout algorithms perform well with respect to their base policy. In this paper, we develop a theoretical approach to provide some answers to this issue in the context of the 0-1 Knapsack Problem, with the hope that this theory may have relevance for more difficult combinatorial problems. We first show how to obtain a minimum performance ratio for the generic rollout algorithm for the 0-1 knapsack problem. Then, we select some of the known heuristic algorithms as base policies. For each of the corresponding rollout algorithms, we prove a specific minimum performance ratio and the worst-case performance ratio.

The paper is organized as follows. In Sect. 2, we recall the 0-1 knapsack problem, some of the classical heuristic algorithms proposed for its solution and some variants.

For each of these algorithms, we recall the worst-case performance ratio, if known. Otherwise, we prove it. In Sect. 3, we describe the generic rollout algorithm for the solution of the 0-1 knapsack problem. In Sect. 4, we show the minimum performance ratio for the generic rollout algorithm and, then, specific minimum performance ratios for specific rollout algorithms. Finally, in Sect. 5, we prove the worst-case performance ratio of these rollout algorithms.

2 The 0-1 Knapsack Problem

The 0-1 knapsack problem is one of the most studied combinatorial optimization problems. A set U of n items and a knapsack of capacity b are available. Each item $i \in U$ has a positive weight w_i, with $w_i \leq b$, and a positive profit p_i. The aim is to find a subset of U that maximizes the total profit and satisfies the capacity of the knapsack. Let U^* and $z^* = \sum_{i \in U^*} p_i$ be an optimal subset of U and the corresponding profit, respectively. Very effective exact algorithms are known as well as several heuristics, approximation algorithms and schemes (we refer to [14] and [15]).

Let us recall the concept of performance ratio and worst-case performance ratio of a heuristic algorithm. Consider a maximization problem. Let $\mathcal{I}$ be the set of instances of this problem, $I \in \mathcal{I}$ a given instance, $z^A(I)$ be the profit obtained by applying an algorithm A to the instance I, and $z^*(I)$ be the corresponding optimal profit. The performance ratio of the algorithm A in the instance I is the ratio between $z^A(I)$ and $z^*(I)$, while the worst-case performance ratio of the algorithm A is:

$$W(A) = inf_{I \in \mathcal{I}} \left\{ \frac{z^A(I)}{z^*(I)} \right\}.$$

Obviously, $0 \leq W(A) \leq 1$ and the closer $W(A)$ is to one, the better is the worst-case performance of A. To prove that $W(A) = \rho$, for some ρ, two phases have to be carried out. In the first, we have to prove that $\frac{z^A(I)}{z^*(I)} \geq \rho$ for any instance $I \in \mathcal{I}$. In the second, we have to prove either that there exists a specific instance $I' \in \mathcal{I}$ such that $\frac{z^A(I')}{z^*(I')} = \rho$ or that there exists a series of instances for which the performance ratio tends to be arbitrarily close to ρ. In this case, the worst-case performance ratio is tight. In the reminder of the paper, for the sake of simplicity, we will omit the reference to the instance I.

We consider the following heuristics as base policies of rollout algorithms.

The p_i-Greedy Algorithm The items are inserted on the basis of the non-increasing order of p_i. Following this ordering, each item is inserted if its weight is not greater than the residual capacity. Let z^{PG} be the profit obtained by applying this algorithm. The following instance shows that $\frac{z^{PG}}{z^*} \to 0$. Instance: $n = 1 + 1/\epsilon$, where $\epsilon \to 0$, $p_1 = w_1 = 1$ and $p_i = 1 - \epsilon$ and $w_i = \epsilon$, respectively, for the remaining $1/\epsilon$ items. Note that this instance can be also used to prove that the algorithm in which the item with maximum profit only is inserted in the knapsack has a worst-case performance ratio which tends to 0.

The Greedy Algorithm First the items are ordered in the non-increasing order of p_i/w_i and then, following this ordering, the items are inserted in the knapsack, until the first item for which the capacity is not satisfied if inserted. It is well known that the *Greedy Algorithm* has a tight worst-case performance ratio of 0 (see [14] and [15]).

The Improved Greedy Algorithm It is an improved version of the *Greedy Algorithm* in which the items are ordered in the non-increasing order of p_i/w_i and then, following this ordering, each item is inserted in the knapsack if its weight is not greater than the residual capacity. We now prove that this algorithm has the same worst-case performance of the *Greedy Algorithm*. Let z^{IG} be the profit obtained by applying this algorithm. The following instance shows that $\frac{z^{IG}}{z^*} \to 0$. Instance: $n = 2$, $p_1 = 2$, $p_2 = M$, $w_1 = 1$ and $w_2 = M$, where $M \geq 2$, and capacity $b = M$.

The Ext-Greedy Algorithm The solution given by this algorithm is the best between the solution of the *Greedy Algorithm* and the solution obtained by inserting in the knapsack the item with maximum profit only. This algorithm has a well known tight worst-case performance ratio of $1/2$ (see [14] and [15]).

The Improved Ext-Greedy Algorithm The solution given by this algorithm is the best between the following two solutions. The first is the one obtained by applying the *Improved Greedy Algorithm*. The second is obtained by applying the p_i*-Greedy Algorithm*. We prove that the *Improved Ext-Greedy Algorithm* has the same worst-case performance ratio of the *Ext-Greedy Algorithm*. Let z^{IEG} be the profit obtained by applying this algorithm. Then, since $\frac{z^{IEG}}{z^*} \geq \frac{z^{EG}}{z^*}$, $\frac{z^{IEG}}{z^*} \geq \frac{1}{2}$. The following instance shows that this bound is tight. Instance: $n = 4$, profits and weights

i	1	2	3	4
p_i	2	M	M	$M + \frac{1}{M}$
w_i	1	M	M	$2M$

where $M \geq 2$, and capacity $b = 2M$.

These algorithms are easy to be implemented and require low computational times. For the sake of completeness, we recall that several Polynomial Time Approximation Schemes (PTASs) and Fully Polynomial Time Approximation Schemes (FPTASs) are also known. Any PTAS for the knapsack problem is based on the idea of guessing a set of items included in the optimal solution by going through all possible candidate sets and then filling the remaining capacity by applying a greedy algorithm. We recall the classical PTAS in [16] and the PTAS in [17], which improves the PTAS by [16] in terms of running time. Any FPTAS is based on the idea of scaling the profit values and then applying dynamic programming on the resulting instance. We recall the earliest FPTAS in [18], the classical FPTAS in [19] and the best known FPTAS in [20] and [21].

3 Rollout Algorithms for the 0-1 Knapsack Problem

In this section, we describe the generic rollout algorithm for the 0-1 Knapsack Problem. Let A_k be a heuristic algorithm for the 0-1 Knapsack Problem with worst-case performance ratio $0 \leq k < 1$, $z^{A_k}(S, c)$ be the profit obtained by applying the algorithm A_k on the set $S \subseteq U$ when the residual capacity is c. The corresponding rollout algorithm $A_k R$, with total profit $z^{A_k R}$, can be described as follows.

Algorithm $A_k R$

1. Set $S := I$, $c := b$ and $z^{A_k R}$:=0.
2. Determine the set $\bar{S} = \{i \in S : w_i \leq c\}$. If $\bar{S} = \emptyset$, then stop. Otherwise:
   (a) For each item $i \in \bar{S}$, compute the estimated total profit obtained by inserting i in the knapsack and then applying A_k on the residual set of items $\bar{S} \backslash \{i\}$ and on the residual capacity $c - w_i$, that is,

$$\Pi_i = p_i + z^{A_k}(\bar{S} \backslash \{i\}, c - w_i).$$

   (b) Select the item $\hat{i}$ such that $\Pi_{\hat{i}} = \max_{j \in \bar{S}} \Pi_j$.
   (c) Set $S := S \backslash \{\hat{i}\}$, $c := c - w_{\hat{i}}$ and $z^{A_k R} := z^{A_k R} + p_{\hat{i}}$ and go to 2.

4 Minimum Performance Ratios

In this section, we aim at answering to our first question. Suppose that the base policy of a rollout algorithm has a performance ratio in a given instance equal to α, where $0 \leq \alpha \leq 1$. The question is: Is it possible to prove a minimum performance ratio for the corresponding rollout algorithm in the same instance? The minimum performance ratios that we now prove are obtained by running only the first iteration of the corresponding rollout algorithm, and therefore with a computational time which is at most n times the one of the corresponding base policy. Our aim is to show that, even if only one iteration is run, rollout algorithms are able to significantly improve the performance of the corresponding base policy. Moreover, we show that, for the generic rollout algorithm and for all the specific rollout algorithms we study, there exists an instance in which the worst-case performance ratio of the rollout algorithm is obtained at the first iteration. Therefore, in the worst-case, no further improvement can be obtained by running more than one iteration.

4.1 A Minimum Performance Ratio for the Generic Rollout Algorithm

We now prove a minimum performance ratio obtained by applying the generic rollout algorithm $A_k R$ for any instance, such that $\frac{z^{A_k}}{z^*} = \alpha$, with $k \leq \alpha \leq 1$. Let U^{A_k} be the set of items selected by the algorithm A_k, U^* be the set of items in the optimal solution and $\hat{U}^* = \{i \in U^* : i \notin U^{A_k}\}$. We denote by n^* the cardinality of the set $\hat{U}^*$, by j the item in the set $\hat{U}^*$ with maximum profit, by $U_j = U - \{j\}$ the set of all items but item j and by $b_j = b - w_j$ the residual capacity when the item j is inserted in the knapsack. Finally, we denote by $z^*(S, c)$ the optimal profit on a given set $S \subseteq U$ when the capacity is c.

Theorem 4.1 *For any instance such that* $\frac{z^{A_k}}{z^*} = \alpha$, *then* $\frac{z^{A_k R}}{z^*} \geq \max\{\alpha, \frac{1-\alpha}{n}(1-k)+k\}$.

Proof If $z^* = p_j$, then $z^{A_k R} = z^*$. Otherwise, since $z^* = p_j + z^*(U_j, b_j)$ and $\frac{z^{A_k}(U_j,b_j)}{z^*(U_j,b_j)} \geq k$, then

$$\frac{z^{A_k R}}{z^*} \geq \max\left\{\frac{z^{A_k}}{z^*}, \frac{p_j + z^{A_k}(U_j, b_j)}{p_j + z^*(U_j, b_j)}\right\} \geq \max\left\{\alpha, \frac{p_j + kz^*(U_j, b_j)}{p_j + z^*(U_j, b_j)}\right\}.$$

Let us define $0 < \beta < 1$ such that $p_j = \beta z^*$. Since $z^* = p_j + z^*(U_j, b_j)$, then $p_j = \beta(p_j + z^*(U_j, b_j))$, that is, $p_j = \frac{\beta}{1-\beta} z^*(U_j, b_j)$. Therefore,

$$\frac{z^{A_k R}}{z^*} \geq \max\left\{\alpha, \frac{\frac{\beta}{1-\beta} z^*(U_j, b_j) + kz^*(U_j, b_j)}{\frac{\beta}{1-\beta} z^*(U_j, b_j) + z^*(U_j, b_j)}\right\} = \max\{\alpha, \beta(1-k)+k\}.$$

Since $\frac{z^{A_k}}{z^*} = \alpha$, then $\sum_{i \in \hat{U}^*} p_i \geq (1-\alpha)z^*$. Therefore, $p_j \geq \frac{\sum_{i \in \hat{U}^*} p_i}{n^*} \geq \frac{(1-\alpha)z^*}{n^*}$ and $\beta \geq \frac{(1-\alpha)}{n^*}$.

$$\frac{z^{A_k R}}{z^*} \geq \max\left\{\alpha, \frac{1-\alpha}{n^*}(1-k)+k\right\} \geq \max\left\{\alpha, \frac{1-\alpha}{n}(1-k)+k\right\}.$$

□

Note that in the worst case, when $k = 0$, $\alpha \to 0$ and $n \to \infty$, this minimum performance ratio tends to 0.

To better understand the values of the minimum performance ratio of the $A_k R$ algorithm, we consider the case with $k = 0$ and the case with $k = 1/2$ and show the value of the minimum performance ratio for different values of α and n.

Let us first consider the case with $k = 0$. The following corollary shows that for $\alpha \geq \frac{1}{2}$, the corresponding rollout algorithm $A_0 R$ has a performance identical to the one of its base policy.

Corollary 4.1 *For* $k = 0$ *and* $\alpha \geq \frac{1}{2}$, $\max\{\alpha, \frac{1-\alpha}{n}(1-k)+k\} = \alpha, \quad \forall n \geq 1$.

Proof

$$\frac{1-\alpha}{n}(1-k)+k = \frac{1-\alpha}{n} \leq \frac{1-(1/2)}{n} \leq \frac{1}{2} \leq \alpha, \quad \forall n \geq 1.$$

□

Figure 1 shows the value of the minimum performance ratio of the $A_0 R$ (on the y-axis) for different values of α (on the x-axis) and n, when $k = 0$.

The results show that the $A_0 R$ algorithm is able to significantly improve the performance of the A_0 algorithm, especially when A_0 does not perform well, that is, when α is small and when n is small.

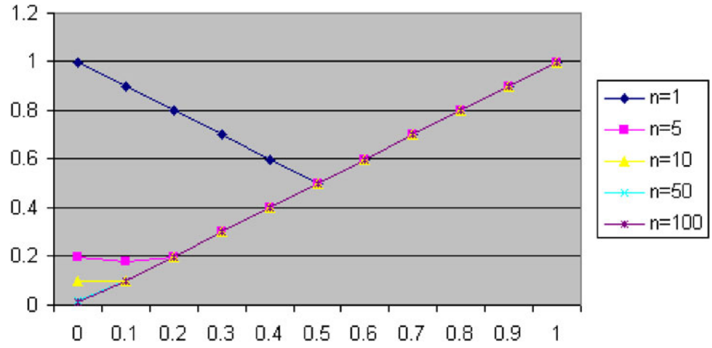

Fig. 1 Value of the minimum performance ratio of the A_0R

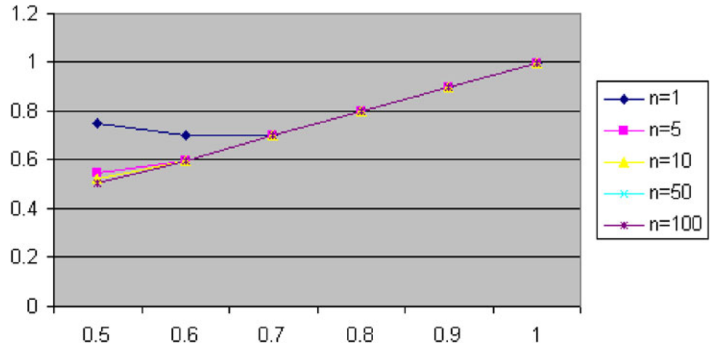

Fig. 2 Value of the minimum performance ratio of the $A_{1/2}R$

Let us now consider the case with $k = 1/2$. The following lemma shows that for $\alpha \geq \frac{2}{3}$, the corresponding rollout algorithm $A_{1/2}R$ has a performance identical to the one of its base policy.

Corollary 4.2 *For $k = 1/2$ and $\alpha \geq 2/3$*, $\max\{\alpha, \frac{1-\alpha}{n}(1-k)+k\} = \alpha, \forall n \geq 1$.

Proof

$$\frac{1-\alpha}{n}(1-k)+k = \frac{1-\alpha}{2n} + \frac{1}{2} \leq \frac{1-(2/3)}{2n} + \frac{1}{2} = \frac{1}{6n} + \frac{1}{2} \leq \frac{1}{6} + \frac{1}{2} = \frac{2}{3} \leq \alpha.$$

□

Figure 2 shows the value of the minimum performance ratio on the performance of the $A_{1/2}R$ (on the y-axis) for different values of α (on the x-axis) and n.

The results show that, even in this case, the $A_{1/2}R$ algorithm is able to significantly improve the performance of the $A_{1/2}$ algorithm, especially when the $A_{1/2}$ algorithm does not perform well and when n is small.

The following theorem shows that the minimum performance ratio of the generic rollout algorithm stated in Theorem 4.1 is tight, as there exists an instance such that the rollout algorithm in which the p_i-*Greedy Algorithm* is applied as base policy has performance ratio equal to 0. Let z^{PGR} be the profit obtained by applying this algorithm, referred to as PGR.

Theorem 4.2 *There exists an instance such that* $\frac{z^{PGR}}{z^*} \to 0$.

Proof Consider the following instance: $n = 3 + 2/\epsilon$, where $\epsilon \to 0$. The profits and weights of the first three items are:

i	1	2	3
p_i	7ϵ	6ϵ	3ϵ
w_i	2	$2-\epsilon$	$2-2\epsilon$

while each of the remaining $2/\epsilon$ items has profit 2ϵ and weight ϵ. The capacity of the knapsack is $b = 2$.

Let us apply the PGR algorithm. The following table shows the first iteration. The estimated future profit is the profit obtained by applying the p_i-*Greedy Algorithm* on the residual set of items and capacity. The number(s) in parentheses is(are) the corresponding inserted items.

i	1	2		3		4		5		…	n	
p_i	7ϵ	6ϵ		3ϵ		2ϵ		2ϵ		…	2ϵ	
Estimated future profit	0	2ϵ	(4)	4ϵ	(4, 5)	6ϵ	(2)	6ϵ	(2)	…	6ϵ	(2)
Total estimated profit	7ϵ	8ϵ		7ϵ		8ϵ		8ϵ		…	8ϵ	

In the second iteration, if the item 2 has been inserted in the first iteration, one of the items with profit 2ϵ and weight ϵ is inserted, with a total profit of 8ϵ. Otherwise, if one of the items with profit 2ϵ and weight ϵ, say item 4, has been inserted in the first iteration, we have

i	1	2	3		4	5		…	n	
Profit of the first iteration	2ϵ	2ϵ	2ϵ		–	2ϵ		…	2ϵ	
p_i	–	6ϵ	3ϵ		–	2ϵ		…	2ϵ	
Estimated future profit	–	0	2ϵ	(4)	–	3ϵ	(3)	…	3ϵ	(3)
Total estimated profit	2ϵ	8ϵ	7ϵ		–	7ϵ		…	7ϵ	

Therefore, the profit generated by the PGR algorithm is 8ϵ. The optimal profit is given by the solution in which the $2/\epsilon$ items with profit 2ϵ and weight ϵ are inserted in the knapsack, that is, $z^* = 4$. Therefore, on this instance $\frac{z^{PGR}}{z^*} = \frac{8\epsilon}{4} \to 0$, for $\epsilon \to 0$. □

4.2 Specific Minimum Performance Ratios

We now prove a specific minimum performance ratio for the rollout algorithm based on the *Greedy Algorithm*, referred to as *GR*, and the rollout algorithm based on the

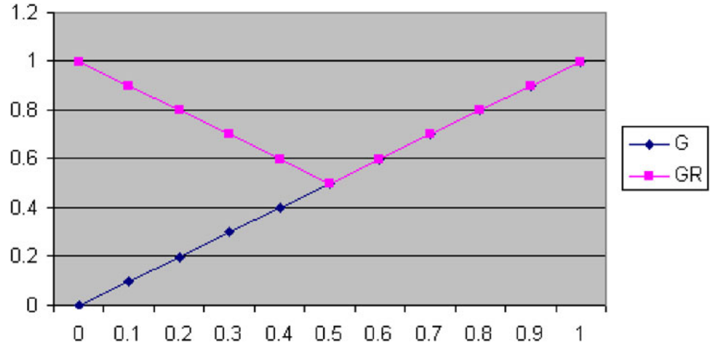

Fig. 3 Value of the minimum performance ratio of the *GR*

Ext-Greedy Algorithm, referred to as *EGR*. Let z^{GR} be the profit of the rollout algorithm based on the *Greedy Algorithm*.

Theorem 4.3 *For any instance such that* $\frac{z^G}{z^*} = \alpha$, *then* $\frac{z^{GR}}{z^*} \geq \max\{\alpha, 1-\alpha\}$.

Proof Consider the LP relaxation of the 0-1 knapsack problem. Assume that the set of items U be ordered on the basis of the ratio p_i/w_i. Let s be the critical item, that is, the item s such that $s = \min(\tau \in U : \sum_{i=1}^{\tau} w_i > b)$. It is well known that $z^{LP} \leq \sum_{i=1}^{s} p_i$ and that the solution obtained by inserting in the knapsack the items $i = 1, 2, \ldots, s-1$ is identical to the solution obtained by the *Greedy Algorithm*. The corresponding profit is $z^G = \sum_{i=1}^{s-1} p_i$. Therefore,

$$p_s \geq z^{LP} - z^G \geq z^* - z^G = z^* - \alpha z^* = (1-\alpha)z^*.$$

Therefore, $\frac{z^{GR}}{z^*} \geq \max\{\frac{z^G}{z^*}, \frac{p_s}{z^*}\} \geq \max\{\alpha, 1-\alpha\}$. □

From Fig. 3, it is easy to see that this minimum performance ratio is equal to 1 for $\alpha = 0$, then it linearly decreases until $\alpha = 1/2$, where it is equal to $1/2$, and then it linearly increases until $\alpha = 1$, where it is again equal to 1. Note that this minimum performance ratio does not depend on the value of n. By comparing the minimum performance of *GR* with the corresponding performance of the *Greedy Algorithm*, we can conclude that the *GR* significantly dominates the *Greedy Algorithm* for $\alpha < 1/2$, in particular when α is small. Obviously, this result holds also for the rollout based on the *Improved Greedy Algorithm*, referred to as *IGR*.

Let z^{EGR} be the profit of the rollout algorithm based on the *Ext-Greedy Algorithm*.

Theorem 4.4 *For any instance such that* $\frac{z^{EG}}{z^*} = \alpha$, $\frac{z^{EGR}}{z^*} \geq \max\{\alpha, \frac{2}{3}\}$.

Proof Let j^* be the item with maximum profit in the optimal solution, U_{j^*} be the set of all items but the item j^*, that is, $U_{j^*} = U - \{j^*\}$, and $b_{j^*} = b - w_j^*$ be the residual capacity when the item j^* is inserted in the knapsack. Finally, recall that we denote by $z^*(S, c)$ and $z^{EG}(S, c)$ the optimal profit and the profit obtained by applying the *Ext-Greedy Algorithm*, respectively, on a given set $S \subseteq U$ when the capacity is c.

We distinguish the following two cases.

Case 1: $p_{j^*} \geq \frac{1}{3}z^*$

$$\frac{z^{EGR}}{z^*} \geq \max\left\{\alpha, \frac{p_{j^*} + z^{EG}(U_{j^*}, b_{j^*})}{p_{j^*} + z^*(U_{j^*}, b_{j^*})}\right\} \geq \max\left\{\alpha, \frac{p_{j^*} + \frac{1}{2}z^*(U_{j^*}, b_{j^*})}{p_{j^*} + z^*(U_{j^*}, b_{j^*})}\right\}.$$

Let us define $\beta \geq \frac{1}{3}$ such that $p_{j^*} = \beta z^*$.

Since $z^* = p_{j^*} + z^*(U_{j^*}, b_{j^*})$, then $p_{j^*} = \beta(p_{j^*} + z^*(U_{j^*}, b_{j^*}))$, that is, $p_{j^*} = \frac{\beta}{1-\beta}z^*(U_{j^*}, b_{j^*})$. Therefore,

$$\begin{aligned}\frac{z^{EGR}}{z^*} &\geq \max\left\{\alpha, \frac{\frac{\beta}{1-\beta}z^*(U_{j^*}, b_{j^*}) + \frac{1}{2}z^*(U_{j^*}, b_{j^*})}{\frac{\beta}{1-\beta}z^*(U_{j^*}, b_{j^*}) + z^*(U_{j^*}, b_{j^*})}\right\} \\ &= \max\left\{\alpha, \frac{\beta+1}{2}\right\} \geq \max\left\{\alpha, \frac{2}{3}\right\}.\end{aligned}$$

Case 2: $p_{j^*} < \frac{1}{3}z^*$ Assume that the set of items U be ordered on the basis of the ratio $\frac{p_i}{w_i}$ and that the item j^* be inserted in the knapsack by the *EGR* algorithm. Let $U_{j^*}^G$ be the set of items selected by applying the *Greedy Algorithm* to the residual capacity and items. Let k^* be the first item in the optimal solution not selected by the *Greedy Algorithm*, $U_{j^*}^I$ and $U_{j^*}^N$ be the set of items in the optimal solution selected and not selected, respectively, by the *Greedy Algorithm* and $U_{j^*}^{GN}$ the set of items selected by the *Greedy Algorithm* but not in the optimal solution. Then,

$$z^* = p_{j^*} + \sum_{i \in U_{j^*}^I} p_i + \sum_{i \in U_{j^*}^N} p_i \leq p_{j^*} + \sum_{i \in U_{j^*}^I} p_i + \frac{p_{k^*}}{w_{k^*}}\left(b - w_{j^*} - \sum_{i \in U_{j^*}^I} w_i\right).$$

Now, the total volume $\sum_{i \in U_{j^*}^{GN}} w_i$ of the items selected by the *Greedy Algorithm*, but not in the optimal solution is not lower than $b - w_{j^*} - \sum_{i \in U_{j^*}^I} w_i - w_k^*$. Moreover, each item in this set has a unit profit not lower than $\frac{p_{k^*}}{w_{k^*}}$. Therefore,

$$\frac{p_{k^*}}{w_{k^*}}\left(b - w_{j^*} - \sum_{i \in U_{j^*}^I} w_i\right) \leq \frac{p_{k^*}}{w_{k^*}} \sum_{i \in U_{j^*}^{GN}} w_i + w_{k^*}\frac{p_{k^*}}{w_{k^*}} \leq \sum_{i \in U_{j^*}^{GN}} p_i + p_{k^*}.$$

Therefore,

$$z^* \leq p_{j^*} + \sum_{i \in U_{j^*}^I} p_i + \sum_{i \in U_{j^*}^{GN}} p_i + p_{k^*} < p_{j^*} + z^G(U_{j^*}, b_{j^*}) + \frac{1}{3}z^*,$$

as $p_{k^*} \leq p_{j^*} < \frac{1}{3}z^*$. Therefore, $z^* \leq \frac{3}{2}(p_{j^*} + z^G(U_{j^*}, b_{j^*}))$.

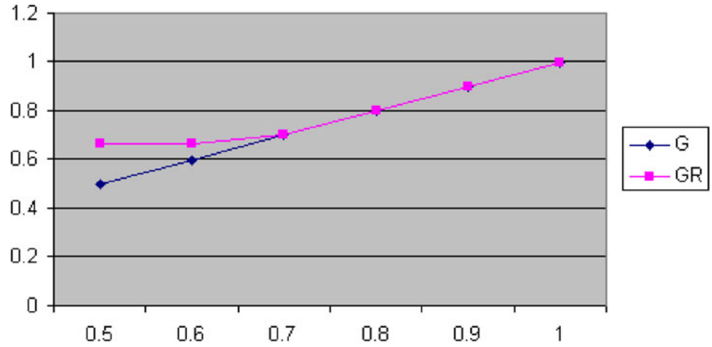

Fig. 4 Value of the minimum performance ratio of the *EGR*

Since $z^{EGR} \geq \max\{\alpha z^*, p_{j^*} + z^{EG}(U_{j^*}, b_{j^*})\} \geq \max\{\alpha z^*, p_{j^*} + z^G(U_{j^*}, b_{j^*})\}$, then

$$\frac{z^{EGR}}{z^*} \geq \max\left\{\alpha, \frac{p_{j^*} + z^G(U_{j^*}, b_{j^*})}{\frac{3}{2}(p_{j^*} + z^G(U_{j^*}, b_{j^*}))}\right\} = \max\left\{\alpha, \frac{2}{3}\right\}. \qquad \square$$

Note that this proof has a structure similar to the one used in [15] to prove the worst-case performance of the PTAS algorithm H^ϵ. However, the proof is different, mainly because, every time one item is considered, the rollout algorithm applies the *Ext-Greedy* algorithm to all the items not already inserted in the knapsack. Instead, every time one item is considered, the algorithm H^ϵ with $l = 1$ applies the *Ext-Greedy* algorithm only to the items not already inserted with profit not greater than the one of the considered item.

From Fig. 4, it is easy to see that this minimum performance ratio is equal to $\frac{2}{3}$ for $\frac{1}{2} \leq \alpha \leq \frac{2}{3}$, then it linearly increases until $\alpha = 1$, where it is equal to 1. By comparing the minimum performance of *EGR* with the corresponding performance of the *Ext-Greedy Algorithm*, we can conclude that the *EGR* significantly dominates the *Ext-Greedy Algorithm*. Obviously, this result also holds for the rollout based on the *Improved Ext-Greedy Algorithm*, referred to as *IEGR*.

5 Worst-Case Performance Ratios

In this section, we provide a worst-case analysis of the rollout algorithms that use, as base policies, the algorithms described and analyzed in Sect. 2. We do not study the rollout algorithm based on the p_i-*Greedy Algorithm*, as we have already shown in the previous section that its worst-case performance ratio is 0.

5.1 The Rollout Algorithm Based on the Greedy Algorithm

We now show that, while the *Greedy Algorithm* has a worst-case performance ratio of 0, the *GR* rollout algorithm has a tight worst-case performance ratio of $1/2$.

Theorem 5.1 $\frac{z^{GR}}{z^*} \geq \frac{1}{2}$ *and the ratio is tight.*

Proof The performance ratio holds thanks to Theorem 4.3, as $\max\{\alpha, 1-\alpha\} \geq \frac{1}{2}$ for $0 \leq \alpha \leq 1$. We now show that this ratio is tight. Consider the following instance: $n = 4$, profits and weights

i	1	2	3	4
p_i	2ϵ	$1+2\epsilon$	1	1
w_i	ϵ	$1+\epsilon$	1	1

where $\epsilon \to 0$, and capacity $b = 2$.

The following table shows the first iteration of the GR algorithm.

i	1		2		3		4	
p_i	2ϵ		$1+2\epsilon$		1		1	
Estimated future profit	$1+2\epsilon$	(2)	2ϵ	(1)	2ϵ	(1)	2ϵ	(1)
Total estimated profit	$1+4\epsilon$		$1+4\epsilon$		$1+2\epsilon$		$1+2\epsilon$	

In the second iteration, if item 1 has been inserted in the first iteration, then item 2 is inserted with a total profit of $1 + 4\epsilon$. Instead, if item 2 has been inserted in the first iteration, then item 1 is inserted with a total profit of $1 + 4\epsilon$. Therefore, the profit generated by the GR algorithm is $z^{GR} = 1 + 4\epsilon$. The optimal profit is given by the solution in which the items 3 and 4 are inserted in the knapsack, that is, $z^* = 2$. Therefore, on this instance $\frac{z^{GR}}{z^*} = \frac{1+4\epsilon}{2} \to \frac{1}{2}$, for $\epsilon \to 0$. □

5.2 The Rollout Algorithm Based on the Improved Greedy Algorithm

We now consider the improved version of the *Greedy Algorithm*, referred to as *Improved Greedy Algorithm*. The items are ordered in the non-increasing order of p_i/w_i and then, following this ordering, each item is inserted in the knapsack if its weight is not greater than the residual capacity. We now show that, while the *Improved Greedy Algorithm* has a worst-case performance ratio of 0, the *IGR* rollout algorithm has a tight worst-case performance ratio of 1/2. Let z^{IGR} be the profit obtained by applying this algorithm.

Theorem 5.2 $\frac{z^{IGR}}{z^*} \geq \frac{1}{2}$ *and the ratio is tight.*

Proof Since $z^{IGR} \geq z^{GR}$, then $\frac{z^{IGR}}{z^*} \geq \frac{z^{GR}}{z^*} \geq \frac{1}{2}$. To prove that the ratio is tight, we consider the instance used in the proof of Theorem 5.1. Since the GR and the IGR algorithms give the same solution in this instance, then $\frac{z^{IGR}}{z^*} \to \frac{1}{2}$, for $\epsilon \to 0$. □

5.3 The Rollout Algorithm Based on the Ext-Greedy Algorithm

The simplest known approximation algorithm with worst-case performance ratio greater than 0 is the so called *Ext-Greedy Algorithm*. We now consider the *EGR* rollout algorithm.

Theorem 5.3 $\frac{z^{EGR}}{z^*} \geq \frac{2}{3}$ *and the ratio is tight.*

Proof The performance ratio holds thanks to Theorem 4.4.

To prove that the ratio is tight, consider the following instance: $n = 5$, profits and weights

i	1	2	3	4	5
p_i	2ϵ	$1+\epsilon$	$1+\epsilon$	1	$2+3\epsilon$
w_i	ϵ	1	1	1	$3-\epsilon$

where $\epsilon < 1/4$, and capacity $b = 3$.

Let us now apply the EGR algorithm. The following table shows the first iteration.

i	1		2		3		4		5	
p_i	2ϵ		$1+\epsilon$		$1+\epsilon$		1		$2+3\epsilon$	
Estimated future profit	$2+3\epsilon$	(5)	$1+3\epsilon$	(1, 3)	$1+3\epsilon$	(1, 2)	$1+3\epsilon$	(1, 2)	2ϵ	(1)
Total estimated profit	$2+5\epsilon$		$2+4\epsilon$		$2+4\epsilon$		$2+3\epsilon$		$2+5\epsilon$	

In the second iteration, if the item 1 has been inserted in the first iteration, we have

i	1	2		3		4		5
Profit of the first iteration	–	2ϵ		2ϵ		2ϵ		2ϵ
p_i	–	$1+\epsilon$		$1+\epsilon$		1		$2+3\epsilon$
Estimated future profit	–	$1+\epsilon$	(3)	$1+\epsilon$	(2)	$1+\epsilon$	(2)	0
Total estimated profit	–	$2+4\epsilon$		$2+4\epsilon$		$2+3\epsilon$		$2+5\epsilon$

If the item 5 has been inserted in the first iteration, then the item 1 is inserted in the second iteration with a total profit of $2 + 5\epsilon$ and no residual capacity. Therefore, the profit generated by the EGR algorithm is $z^{EGR} = 2 + 5\epsilon$. The optimal profit is given by the solution in which the items 2, 3 and 4 are inserted in the knapsack, that is, $z^* = 3 + 2\epsilon$. Hence, on this instance $\frac{z^{EGR}}{z^*} = \frac{2+5\epsilon}{3+2\epsilon} \to \frac{2}{3}$, for $\epsilon \to 0$. □

5.4 The Rollout Algorithm Based on the Improved Ext-Greedy Algorithm

We now consider the *IEGR* rollout algorithm. Let z^{IEGR} be the profit obtained by applying this algorithm.

Theorem 5.4 $\frac{z^{IEGR}}{z^*} \geq \frac{2}{3}$ *and the ratio is tight.*

Proof Since $z^{IEGR} \geq z^{EGR}$, then $\frac{z^{IEGR}}{z^*} \geq \frac{z^{EGR}}{z^*} \geq \frac{2}{3}$. To prove that the ratio is tight, consider the following instance: $n = 6$, profits and weights

i	1	2	3	4	5	6
p_i	2ϵ	$1+\epsilon$	$1+\epsilon$	1	$2+3\epsilon$	$1+2\epsilon$
w_i	ϵ	1	1	1	$3-\epsilon$	2

where $\epsilon < 1/4$, and capacity $b = 3$.

Let us now apply the EGR algorithm. The following table shows the first iteration.

i	1	2	3	4	5	6
p_i	2ϵ	$1+\epsilon$	$1+\epsilon$	1	$2+3\epsilon$	$1+2\epsilon$
Estimated future profit	$2+3\epsilon$ (5)	$1+3\epsilon$ (1, 3)	$1+3\epsilon$ (1, 2)	$1+3\epsilon$ (1, 2)	2ϵ (1)	$1+\epsilon$ (2)
Total estimated profit	$2+5\epsilon$	$2+4\epsilon$	$2+4\epsilon$	$2+3\epsilon$	$2+5\epsilon$	$2+3\epsilon$

In the second iteration, if the item 1 has been inserted in the first iteration, we have

i	1	2	3	4	5	6
Profit of the first iteration	–	2ϵ	2ϵ	2ϵ	2ϵ	2ϵ
p_i	–	$1+\epsilon$	$1+\epsilon$	1	$2+3\epsilon$	$1+2\epsilon$
Estimated future profit	–	$1+\epsilon$ (3)	$1+\epsilon$ (2)	$1+\epsilon$ (2)	0	0
Total estimated profit	–	$2+4\epsilon$	$2+4\epsilon$	$2+3\epsilon$	$2+5\epsilon$	$1+4\epsilon$

If the item 5 has been inserted in the first iteration, then the item 1 is inserted in the second iteration with a total profit of $2 + 5\epsilon$ and no residual capacity. Therefore, the profit generated by the IEGR algorithm is $z^{IEGR} = 2 + 5\epsilon$. The optimal profit is given by the solution in which the items 2, 3 and 4 are inserted in the knapsack, that is, $z^* = 3 + 2\epsilon$. Therefore, on this instance $\frac{z^{IEGR}}{z^*} = \frac{2+5\epsilon}{3+2\epsilon} \rightarrow \frac{2}{3}$ for $\epsilon \rightarrow 0$. □

6 Conclusions

The analysis carried out in this paper is the first theoretical analysis proposed to evaluate the performance of rollout algorithms given the performance of the corresponding base policy. An 'easy' classical combinatorial problem, the 0-1 Knapsack Problem, has been used to introduce the methodology. This analysis allows us to state that rollout algorithms for the 0-1 knapsack problem are able to significantly improve the performance of the corresponding base policies. In particular, the worst-case performance ratios have been improved from 0 to $1/2$ for the *Greedy Algorithm* and the *Improved Greedy Algorithm* and from $1/2$ to $2/3$ for the *Ext-Greedy Algorithm* and the *Improved Ext-Greedy Algorithm*. We also showed that, for each of these rollout algorithms, there exists an instance in which the worst-case performance ratio is obtained at the first iteration. This is interesting because the worst-case performance of the rollout algorithm is guaranteed even if an algorithm with time complexity only n times greater than the one of the base policy is run. Moreover, we also showed how it is possible to formally prove a minimum performance ratio of the rollout algorithms for any instance and, therefore, to estimate a minimum improvement of the rollout algorithms with respect to their base policies. It has been shown that, even if only the first iteration of the rollout algorithm is run, the minimum performance ratio of the rollout algorithms tends to be significantly greater when the performance ratio of the corresponding base policy is poor. Therefore, rollout algorithms are very appealing and tend to significantly improve the average performance of their base policies. Future research can be devoted to quantify the improvement of the minimum performance ratio when more than one iteration is run.

References

1. Bertsekas, D.P., Tsitsiklis, J.N.: Neuro-Dynamic Programming. Athena Scientific, Belmont (1996)
2. Tesauro, G., Galperin, G.R.: On-line policy improvement using Monte Carlo search. Adv. Neural Inf. Process. Syst. **9**, 1068–1074 (1997)
3. Bertsekas, D.P., Castanõn, D.A.: Rollout algorithms for stochastic scheduling problems. J. Heuristics **5**, 89–108 (1998)
4. Secomandi, N.: Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands. Comput. Oper. Res. **27**, 1201–1225 (2000)
5. Secomandi, N.: A rollout policy for the vehicle routing problem with stochastic demands. Oper. Res. **49**, 796–802 (2001)
6. Secomandi, N.: Analysis of a rollout approach to sequencing problems with stochastic routing applications. J. Heuristics **9**, 321–352 (2003)
7. Novoa, C., Storer, R.: An approximate dynamic programming approach for the vehicle routing problem with stochastic demands. Eur. J. Oper. Res. **196**, 509–515 (2009)
8. Guerriero, F., Mancini, M.: A cooperative parallel rollout algorithm for the sequential ordering problem. Parallel Comput. **29**, 663–677 (2003)
9. Guerriero, F., Mancini, M.: Parallelization strategies for rollout algorithms. Comput. Optim. Appl. **31**, 221–244 (2005)
10. Bertsekas, D.P., Tsitsiklis, J.N., Wu, C.: Rollout algorithms for combinatorial optimization. J. Heuristics **3**, 245–262 (1997)
11. Guerriero, F.: Hybrid rollout approaches for the job shop scheduling problem. J. Optim. Theory Appl. **139**, 419–438 (2008)
12. Bertsimas, D.J., Demir, R.: An approximate dynamic programming approach to multidimensional Knapsack problems. Manag. Sci. **48**, 550–565 (2002)
13. Bertsekas, D.P.: Dynamic Programming and Optimal Control, vol. I. Athena Scientific, Belmont (2005)
14. Martello, S., Toth, P.: Knapsack Problems: Algorithms and Computer Implementations. Wiley, Chichester (1990)
15. Kellerer, H., Pferschy, U., Pisinger, D.: Knapsack Problems. Springer, Berlin (2004)
16. Sahni, S.: Approximate algorithms for the 0-1 knapsack problem. J. ACM **22**, 115–124 (1975)
17. Caprara, A., Kellerer, H., Pferschy, U., Pisinger, D.: Approximation algorithms for Knapsack problems with cardinality constraints. Eur. J. Oper. Res. **123**, 333–345 (2000)

18. Ibarra, O.H., Kim, C.E.: Fast approximation algorithms for the knapsack and sum of subset problem. J. ACM **22**, 463–468 (1975)
19. Lawler, E.L.: Fast approximation algorithms for knapsack problems. Math. Oper. Res. **4**, 339–356 (1979)
20. Kellerer, H., Pferschy, U.: A new fully polynomial time approximation scheme for the knapsack problem. J. Comb. Optim. **3**, 59–71 (1999)
21. Kellerer, H., Pferschy, U.: Improved dynamic programming in connection with an FPTAS for the Knapsack problem. J. Comb. Optim. **8**, 5–11 (2004)