Available online at www.sciencedirect.com
SCIENCE DIRECT
ELSEVIER
Discrete Applied Mathematics 154 (2006) 1685–1692
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

Approximation algorithms for the bi-criteria weighted MAX-CUT problem☆

Eric Angel, Evripidis Bampis, Laurent Gourvès

LaMI, CNRS UMR 8042, Université d'Évry Val d'Essonne, France

Received 21 September 2005; received in revised form 30 December 2005; accepted 21 February 2006
Available online 11 May 2006

Abstract

We consider a generalization of the classical MAX-CUT problem where two objective functions are simultaneously considered. We derive some theorems on the existence and the non-existence of feasible cuts that are at the same time near optimal for both criteria. Furthermore, two approximation algorithms with performance guarantee are presented. The first one is deterministic while the second one is randomized. A generalization of these results is given for the bi-criteria MAX-k-CUT problem.

Keywords: Multicriteria optimization; Bicriteria MAX-CUT problem; Approximation algorithm; Performance guarantee

1. Introduction

Given an undirected graph $G=(V,E)$ with non-negative edge weights w_{ij}, the objective of the Maximum Cut problem (MAX-CUT) is to find a partition of the vertex set into two subsets S_1 and S_2, such that the sum of the weights of the edges having endpoints in different subsets is maximum. Formally, the weight of the cut (S_1, S_2) to be maximized is given by

$$W(S_1, S_2) = \sum_{i\in S_1, j\in S_2} w_{ij}.$$

This well known combinatorial problem was shown to be **NP**-complete by Karp [12]. It has applications in many fields including VLSI circuit design and Statistical Physics [5].

In this article, we study a *bi-criteria* version of the MAX-CUT problem. Formally, we are given an undirected graph $G=(V,E)$ and two distinct weighting functions. Each feasible cut is then evaluated with respect to these two criteria.

In general no feasible solution can meet optimality simultaneously for both criteria. However, a set of solutions which *dominate*[1] all the others (the so-called *Pareto curve*) always exists. Because of the complexity of the classical (mono-criterion) MAX-CUT problem, determining this Pareto curve is computationally problematic. Indeed, the bi-criteria

☆ A preliminary version of this article was presented at the 31st Workshop on Graph Theoretic Concepts in Computer Science, WG'2005, June 23–25, Metz, France.

E-mail addresses: angel@lami.univ-evry.fr (E. Angel), bampis@lami.univ-evry.fr (E. Bampis), lgourves@lami.univ-evry.fr (L. Gourvès).

[1] A solution x dominates another solution y if x is at least as good as y for all criteria and strictly better for at least one criterion.

0166-218X/$ - see front matter
doi:10.1016/j.dam.2006.02.008

MAX-CUT problem generalizes MAX-CUT. Moreover, the size of the Pareto curve, i.e. the number of non-dominated solutions, may be exponential.

Concerning *multi-criteria optimization* (see [7] for a recent book on the topic), three different approaches are often followed: the *budget approach*, the *Pareto curve approach* and the *simultaneous approach*. In this article we follow the third one.

By taking as a reference an *ideal solution*, namely a (not necessarily feasible) cut which simultaneously maximizes all objective functions, one tries to compute a feasible cut which approximates this ideal solution with a performance guarantee on each criterion.

In this direction, Stein and Wein [15] considered a scheduling problem with two well studied criteria, namely the *makespan* and the *average weighted completion time*. They derived existence and non-existence theorems on schedules that are simultaneously near-optimal with respect to both objective functions. A series of recent papers follow this approach [1–4,14].

In this article, we study a bi-criteria weighted MAX-CUT problem with the simultaneous approach (each objective is a weighting function). Up to our knowledge, this direction was not already investigated but one can mention some related works. Papadimitriou and Yannakakis [13] show that unless $P = NP$, there is no FPTAS for constructing an ε-approximate Pareto curve for the bi-criteria $s - t$ MIN-CUT problem. They follow the Pareto curve approach which consists in computing a set P_ε of feasible solutions such that any feasible solution is approximated by a solution in P_ε with a performance ratio $(1 + \varepsilon)$ on each criterion. Bruglieri et al. [6] study several *k-cardinality minimum cut problems* which consist in computing a minimal weight cut with at least (or exactly) k edges. In fact, they follow a budget approach[2] where the two criteria are the weight and the number of edges that belong to the cut. Jäger and Srivastav consider a different cardinality constraint in [11]. Their problem consists in determining a subset $S \subseteq V$ of k vertices such that the total weight of the edges connecting S and $V \setminus S$ is maximized.

The paper is organized as follows: A formal presentation of the problem is given in Section 2. Sections 3 and 4 are, respectively, devoted to a deterministic and a randomized bi-criteria approximation algorithm with performance guarantee. The results are extended to the bi-criteria MAX-k-CUT problem[3] (a solution is a partition of V into $k \geqslant 2$ subsets) in Section 5. Finally, some outlooks and concluding remarks are given in Section 6.

2. Formalization and notation

We are given an undirected graph $G = (V, E)$ where each edge $e \in E$ has a non-negative weight w_e and a non-negative length l_e. A solution (S_1, S_2) is feasible if it constitutes a partition of V. An edge e *belongs* to a cut (S_1, S_2), denoted by $e \in (S_1, S_2)$, if e connects a vertex in S_1 and a vertex in S_2. The following objective functions, namely the *total weight* W and the *total length* L, are considered:

$$W(S_1, S_2) = \sum_{e \in (S_1, S_2)} w_e \quad \text{and} \quad L(S_1, S_2) = \sum_{e \in (S_1, S_2)} l_e.$$

Let *OPTW* (resp. *OPTL*) be the maximum total weight (resp. total length) of a feasible cut. A feasible (α, β)-*approximate* cut (A_1, A_2) is such that

$$W(A_1, A_2) \geqslant \alpha\, OPTW \quad \text{and} \quad L(A_1, A_2) \geqslant \beta\, OPTL,$$

where $0 < \alpha \leqslant 1$ and $0 < \beta \leqslant 1$. An (α, β)-*approximation algorithm* outputs a solution which is simultaneously α-approximate on the first criterion (the total weight) and β-approximate on the second criterion (the total length).

3. A deterministic approximation algorithm

Given a deterministic α-approximation algorithm **Al** for the mono-criterion weighted MAX-CUT problem, one can build an $(\alpha/2, \alpha/2)$-approximation algorithm for the bi-criteria weighted MAX-CUT problem. The algorithm called **Bi-Approx** is given in Table 1.

[2] One tries to optimize a criterion while the other is constrained by a budget (k here).

[3] In some literature like [11] MAX-k-CUT denotes a different problem with a cardinality constraint.

Table 1
A deterministic approximation algorithm for the bi-criteria MAX-CUT problem

Bi-Approx	
Input:	G and **Al**
Step 1:	Find (R_1, R_2) with **Al** such that $W(R_1, R_2) \geqslant \alpha.OPTW$
Step 2:	Find (S_1, S_2) with **Al** such that $L(S_1, S_2) \geqslant \alpha.OPTL$
Step 3:	Build (T_1, T_2) such that $T_1 = (R_1 \cap S_1) \cup (R_2 \cap S_2)$
Step 4:	*If* $L(R_1, R_2) \geqslant 0.5\, L(S_1, S_2)$
	Then Return (R_1, R_2)
	Else If $W(S_1, S_2) \geqslant 0.5\, W(R_1, R_2)$
	Then Return (S_1, S_2)
	Else Return (T_1, T_2)

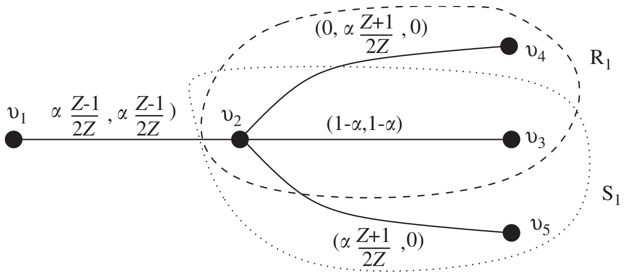

Fig. 1. Instance for which **Bi-Approx** returns an $(\alpha/2, \alpha/2)$-approximate cut (T_1, T_2) where $T_1 = \{v_1, v_2, v_3\}$.

Theorem 1. **Bi-Approx** *is a deterministic* $(\alpha/2, \alpha/2)$-*approximation algorithm for the bi-criteria weighted* MAX-CUT *problem if* **Al** *is a deterministic* α-*approximation algorithm for the mono-criterion weighted* MAX-CUT *problem.*

Proof. Let E_1, E_2 and E_3 be three subsets of E defined as follows:

$$E_1 = \{e \in E \mid e \in (R_1, R_2) \text{ and } e \notin (S_1, S_2)\},$$
$$E_2 = \{e \in E \mid e \notin (R_1, R_2) \text{ and } e \in (S_1, S_2)\},$$
$$E_3 = \{e \in E \mid e \in (R_1, R_2) \text{ and } e \in (S_1, S_2)\}.$$

We have $(R_1, R_2) = E_1 \cup E_3$, $(S_1, S_2) = E_2 \cup E_3$ and $(T_1, T_2) = E_1 \cup E_2$.

If $\sum_{e\in E_3} l_e \geqslant \sum_{e\in E_2} l_e$ then $L(R_1, R_2) \geqslant 0.5L(S_1, S_2)$. Thus, **Bi-Approx** returns (R_1, R_2) which is $(\alpha, \alpha/2)$-approximate and hence $(\alpha/2, \alpha/2)$-approximate. If $\sum_{e\in E_3} w_e \geqslant \sum_{e\in E_1} w_e$ then $W(S_1, S_2) \geqslant 0.5W(R_1, R_2)$. Thus, **Bi-Approx** returns (S_1, S_2) which is $(\alpha/2, \alpha)$-approximate and hence $(\alpha/2, \alpha/2)$-approximate. If $\sum_{e\in E_3} w_e < \sum_{e\in E_1} w_e$ and $\sum_{e\in E_3} l_e < \sum_{e\in E_2} l_e$ then

$$W(T_1, T_2) \geqslant 0.5W(R_1, R_2) \quad \text{and} \quad L(T_1, T_2) \geqslant 0.5L(S_1, S_2).$$

Thus, **Bi-Approx** returns (T_1, T_2) which is $(\alpha/2, \alpha/2)$-approximate. □

The analysis of **Bi-Approx** is asymptotically tight. To see it, consider the instance given in Fig. 1 where Z is a large integer. The cut (I_1, I_2) such that $I_1 = \{v_2\}$ is optimal for the two criteria, its total weight and total length are both equal to 1. The cut (R_1, R_2) achieves the values $(\alpha, \alpha(Z-1)/2Z)$ while (S_1, S_2) achieves the values $(\alpha(Z-1)/2Z, \alpha)$. Since $\alpha(Z-1)/2Z < 0.5\alpha$, **Bi-Approx** returns a solution (T_1, T_2) such that $T_1 = \{v_1, v_2, v_3\}$ and its total weight and total length are both equal to $\alpha(Z+1)/2Z$. When Z tends to infinity, the solution returned tends to be $(\alpha/2, \alpha/2)$-approximate.

Corollary 1. *There exists a deterministic* $(0.43928, 0.43928)$-*approximation algorithm for the bi-criteria weighted* MAX-CUT *problem.*

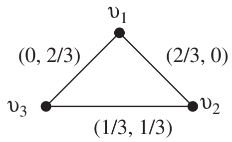

Fig. 2. $OPTW = OPTL = 1$.

Proof. Replace **Al** in **Bi-Approx** by the derandomized algorithm of Goemans and Williamson [8,9] which is a 0.87856-approximation algorithm and the result follows. □

Interestingly, an existence result can be derived from the algorithm **Bi-Approx**.

Corollary 2. *For all instances of the bi-criteria weighted* MAX-CUT *problem, there always exists a feasible* $(\frac{1}{2}, \frac{1}{2})$*-approximate solution.*

Proof. Suppose that **Al** in **Bi-Approx** is an optimal (1-approximation) algorithm for the mono-criterion weighted MAX-CUT problem and the result follows. □

One can remark that we cannot guarantee the existence of a feasible cut whose total weight and total length are at the same time strictly more than $\frac{1}{2}$-approximate. Indeed, consider the complete graph K_3 whose edges e, e' and e'' are such that $w_e = l_{e'} = 0$ and $l_e = w_{e'} = w_{e''} = l_{e''} = 1$. The maximum total weight *OPTW* and the maximum total length *OPTL* are both equal to 2 while no feasible cut has a total weight and a total length simultaneously strictly superior to 1.

4. A randomized approximation algorithm

As usual, we consider that a randomized algorithm for a mono-criterion maximization problem is an α-expected approximation algorithm if the expected value (denoted by $E[X]$) of the solution returned is at least α times the value (denoted by *OPT*) of an optimal solution: $E[X] \geqslant \alpha OPT$.

When randomization is considered, the bi-criteria weighted MAX-CUT problem is then to find a feasible cut (A_1, A_2) such that $E[W(A_1, A_2)] \geqslant \alpha OPTW$ and $E[L(A_1, A_2)] \geqslant \beta OPTL$ where $0 < \alpha \leqslant 1$ and $0 < \beta \leqslant 1$.

Proposition 1. *There is no hope to get an* (α, β)*-expected approximation algorithm for the bi-criteria weighted* MAX-CUT *problem with* $\alpha = \beta$ *and* $\alpha > \frac{2}{3}$.

Proof. To see it, consider the example given in Fig. 2 where $OPTW = OPTL = 1$. Four cuts (R_1, R_2), (S_1, S_2), (T_1, T_2) and (U_1, U_2) are feasible with values, respectively, (0, 0), (2/3, 2/3), (1/3, 1), and (1, 1/3). Let **Ran Al** be a randomized algorithm which outputs (R_1, R_2) with a probability p_1, (S_1, S_2) with a probability p_2, (T_1, T_2) with a probability p_3 and (U_1, U_2) with a probability p_4. Obviously, one has $p_1 + p_2 + p_3 + p_4 = 1$. The expected value of the cut (A_1, A_2) output by **Ran Al** is

$$E[W(A_1, A_2)] = \frac{2p_2}{3} + \frac{p_3}{3} + p_4 \quad \text{and} \quad E[L(A_1, A_2)] = \frac{2p_2}{3} + p_3 + \frac{p_4}{3}.$$

The problem is then to find p_1–p_4 such that $E[W(A_1, A_2)] \geqslant \alpha$, $E[L(A_1, A_2)] \geqslant \alpha$ and α is maximized. When $p_1 = p_3 = p_4 = 0$ and $p_2 = 1$, α reaches $\frac{2}{3}$ which is the best possible value. As a consequence, no randomized algorithm can be (α, α)-expected approximate with $\alpha > 2/3$. □

This statement has a consequence in the approximability of the weighted bi-criteria MAX-CUT problem. Indeed, there is no hope to design a deterministic (α, β)-approximation algorithm such that $\alpha + \beta > \frac{4}{3}$. To see it, suppose that we have such an algorithm.[4] One can build two solutions (R_1, R_2) and (S_1, S_2) such that $W(R_1, R_2) \geqslant \alpha OPTW$, $L(R_1, R_2) \geqslant \beta OPTL$,

[4] Because of the symmetry of the problem, an (α, β)-approximation algorithm is also a (β, α)-approximation one.

Table 2
A randomized approximation algorithm for the bi-criteria MAX-CUT problem

Ran Bi-Approx	
Input:	G and **Al**
Step 1:	Find (R_1, R_2) with **Al** such that $W(R_1, R_2) \geqslant \alpha$ *OPTW*
Step 2:	Find (S_1, S_2) with **Al** such that $L(S_1, S_2) \geqslant \alpha$ *OPTL*
Step 3:	Build (T_1, T_2) such that $T_1 = (R_1 \cap S_1) \cup (R_2 \cap S_2)$
Step 4:	Let $\gamma = (3 - \sqrt{5})/2$
Step 5:	*If* $W(S_1, S_2) \geqslant \gamma W(R_1, R_2)$
	Then If $L(R_1, R_2) \geqslant \gamma L(S_1, S_2)$
	Then Return (R_1, R_2) with a probability 0.5
	and (S_1, S_2) with a probability 0.5
	Else Return (R_1, R_2) with a probability γ
	and (S_1, S_2) with a probability $1 - \gamma$
	Else If $L(R_1, R_2) \geqslant \gamma L(S_1, S_2)$
	Then Return (R_1, R_2) with a probability $1 - \gamma$
	and $(S_1, S_2)$with a probability γ
	Else Return (T_1, T_2)

$W(S_1, S_2) \geqslant \beta OPTW$ and $L(S_1, S_2) \geqslant \alpha OPTL$. Now consider the randomized algorithm which consists in returning equiprobably either (R_1, R_2) or (S_1, S_2). We would get a $((\alpha + \beta)/2, (\alpha + \beta)/2)$-expected approximate cut and $(\alpha + \beta)/2 > \frac{2}{3}$.

The algorithm (called **Ransam** in [10]) which consists in building a cut (A_1, A_2) by putting equiprobably a vertex $v \in V$ to either A_1 or A_2 is $\frac{1}{2}$-expected approximate for the mono-criterion weighted MAX-CUT problem. One can remark that it achieves the same performance guarantee for a multi-criteria weighted MAX-CUT problem. However, a better randomized algorithm can be built for the bi-criteria MAX-CUT problem. We propose an algorithm called **Ran Bi-Approx** (see Table 2) which uses a deterministic α-approximation algorithm (called **Al** in the following) for the mono-criterion MAX-CUT problem.

Theorem 2. Ran Bi-Approx *is a randomized* $((\sqrt{5} - 1)/2\alpha, (\sqrt{5} - 1)/2\alpha)$-*expected approximation algorithm for the bi-criteria weighted* MAX-CUT *problem if* **Al** *is a deterministic* α-*approximation algorithm for the mono-criterion weighted* MAX-CUT *problem.*

Proof. Four cases are considered in **Ran Bi-Approx**. For the first one, we suppose that

$$W(S_1, S_2) \geqslant \gamma W(R_1, R_2) \quad \text{and} \quad L(R_1, R_2) \geqslant \gamma L(S_1, S_2).$$

So, we have

$$W(S_1, S_2) \geqslant \gamma \alpha OPTW \quad \text{and} \quad L(R_1, R_2) \geqslant \gamma \alpha OPTL.$$

Since the solution returned in that case is (R_1, R_2) with a probability 0.5 and (S_1, S_2) with a probability 0.5, the expected value on each criterion of the solution returned is at least $\alpha(1 + \gamma)/2$ times the optimum.

For the second case, we suppose that

$$W(S_1, S_2) \geqslant \gamma W(R_1, R_2) \quad \text{and} \quad L(R_1, R_2) \geqslant 0.$$

So, we have $W(S_1, S_2) \geqslant \gamma \alpha OPTW$. Since the solution returned in that case is (R_1, R_2) with a probability $\gamma = (1 - \gamma)/(2 - \gamma)$ and (S_1, S_2) with a probability $1 - \gamma = 1/(2 - \gamma)$, the expected value on each criterion of the solution returned is at least $(1 - \gamma)\alpha$ times the optimum.

The third case is symmetric to the second case, the expected value on each criterion of the solution returned is at least $(1 - \gamma)\alpha$ times the optimum.

For the fourth case, we suppose that

$$W(S_1, S_2) < \gamma W(R_1, R_2), \tag{1}$$

$$L(R_1, R_2) < \gamma L(S_1, S_2). \tag{2}$$

In that case, (T_1, T_2) is returned and its value on each criterion is at least $(1-\gamma)\alpha$ times the optimum. From inequality (1) we get (see the proof of Theorem 1 for a definition of E_1, E_2 and E_3):

$$\sum_{e \in E_2 \cup E_3} w_e < \gamma \sum_{e \in E_1 \cup E_3} w_e,$$

$$(1-\gamma) \sum_{e \in E_1 \cup E_3} w_e + \sum_{e \in E_2} w_e < \sum_{e \in E_1} w_e,$$

$$(1-\gamma) W(R_1, R_2) < (1-\gamma) \sum_{e \in E_1 \cup E_3} w_e + 2 \sum_{e \in E_2} w_e < \sum_{e \in E_1 \cup E_2} w_e = W(T_1, T_2).$$

From inequality (2) we get:

$$\sum_{e \in E_1 \cup E_3} l_e < \gamma \sum_{e \in E_2 \cup E_3} l_e,$$

$$(1-\gamma) \sum_{e \in E_2 \cup E_3} l_e + \sum_{e \in E_1} l_e < \sum_{e \in E_2} l_e,$$

$$(1-\gamma) L(S_1, S_2) < (1-\gamma) \sum_{e \in E_2 \cup E_3} l_e + 2 \sum_{e \in E_1} l_e < \sum_{e \in E_1 \cup E_2} l_e = L(T_1, T_2).$$

Since $(1+\gamma)/2 > 1-\gamma = (\sqrt{5}-1)/2$, the solution returned by **Ran Bi-Approx** has an expected value on each criterion which is at least $(\sqrt{5}-1)/2\alpha$ times the optimum. $\square$

Corollary 3. *There exists a randomized* $(0.54297, 0.54297)$*-expected approximation algorithm for the bi-criteria weighted* MAX-CUT *problem.*

Proof. Replace **Al** by the derandomized algorithm of Goemans and Williamson [8,9] in **Ran Bi-Approx** and the result follows. $\square$

4.1. Discussion

Concerning a possible improvement of the randomized algorithm, one can remark that among the four considered cases, the first one gives a better performance ratio than the others: $(1+\gamma)/2 > 1/(2-\gamma) = (1-\gamma)$ when $\gamma = (3-\sqrt{5})/2$. Then, it is only natural to think that one could improve the result with a slight change of γ and get a situation where the performance ratio is the same for all cases. Unfortunately, no value of γ between 0 and 1 can satisfy $(1+\gamma)/2 = 1/(2-\gamma) = (1-\gamma)$. Moreover, taking a γ different from $(3-\sqrt{5})/2$ leads to a lower approximation ratio. Since we follow a worst case analysis, the performance ratio of the algorithm is $\min\{\alpha(1+\gamma)/2, \alpha/(2-\gamma), \alpha(1-\gamma)\}$. This ratio is maximized when $\gamma = (3-\sqrt{5})/2$.

5. The bi-criteria MAX-*k*-CUT problem

The MAX-*k*-CUT problem is a generalization of MAX-CUT where the node set V is partitioned into k sets $S_1, \ldots, S_k$. The algorithms designed for the bi-criteria MAX-CUT problem and presented in Sections 3 and 4 can be generalized to the bi-criteria MAX-*k*-CUT problem. We still consider that we are given an algorithm called **Al** which returns an α-approximate solution for the mono-criterion MAX-*k*-CUT problem.

Theorem 3. *Given a deterministic* α*-approximation algorithm for the mono-criterion weighted* MAX-*k*-CUT *problem, one can design a deterministic* $(\alpha/2, \alpha/2)$*-approximation algorithm and a randomized* $((\sqrt{5}-1)/2\alpha, (\sqrt{5}-1)/2\alpha)$*-expected approximation algorithm for the bi-criteria weighted* MAX-*k*-CUT *problem.*

Proof. Consider the algorithms **Bi-Approx** and **Ran Bi-Approx** and adapt them as follows:

- replace (R_1, R_2) by $(R_1, \ldots, R_k)$;
- replace (S_1, S_2) by $(S_1, \ldots, S_k)$;
- replace (T_1, T_2) by $(T_1, \ldots, T_k)$ whose new definition is

$$T_{i+1} = \bigcup_{j=0}^{k-1} (R_{((j+i) \bmod k)+1} \cap S_{j+1}), \quad i = 0, \ldots, k-1.$$

Let E_1, E_2 and E_3 be three subsets of E defined as follows:

$$E_1 = \{e \in E \mid e \in (R_1, \ldots, R_k) \text{ and } e \notin (S_1, \ldots, S_k)\},$$
$$E_2 = \{e \in E \mid e \notin (R_1, \ldots, R_k) \text{ and } e \in (S_1, \ldots, S_k)\},$$
$$E_3 = \{e \in E \mid e \in (R_1, \ldots, R_k) \text{ and } e \in (S_1, \ldots, S_k)\}.$$

We have $(R_1, \ldots, R_k) = E_1 \cup E_3$ and $(S_1, \ldots, S_k) = E_2 \cup E_3$. We only prove that $E_1 \cup E_2 \subseteq (T_1, \ldots, T_k)$ because the remaining part of the proof is given in the proofs of Theorems 1 and 2.

Let v and w be two nodes such that the edge $e = (v, w)$ belongs to E_1. Since $e \in (R_1, \ldots, R_k)$, there are two distinct integers a and b such that $v \in R_a$ and $w \in R_b$. Moreover, $e \notin (S_1, \ldots, S_k)$ means that there is an integer c such that v and w both belong to S_c. We have $v \in R_a \cap S_c$ and $w \in R_b \cap S_c$. Now, suppose that $e \notin (T_1, \ldots, T_k)$. So, there is an integer d which satisfies $v \in T_d$ and $w \in T_d$. Since $R_a \neq R_b$, $R_a \cap S_c$ and $R_b \cap S_c$ must appear in the definition of T_d but, by construction, it is impossible. As a consequence, $e \in E_1$ implies $e \in (T_1, \ldots, T_k)$. With similar arguments, we can observe that $e \in E_2$ implies $e \in (T_1, \ldots, T_k)$. Finally $W(T_1, T_2) \geqslant \sum_{e \in E1 \cup E_2} w_e$ and $L(T_1, T_2) \geqslant \sum_{e \in E1 \cup E_2} l_e$. $\square$

One can show that the analysis of the deterministic $(\alpha/2, \alpha/2)$-approximation algorithm for the bi-criteria MAX-k-CUT problem is asymptotically tight. Suppose that $k = 3$ and consider the instance depicted in Fig. 1. The feasible cut $(I_1, I_2, I_3) = (\{v_2\}, \{v_1, v_4\}, \{v_3, v_5\})$ is simultaneously optimal for the total weight and the total length ($OPTW = OPTL = 1$). The cut $(R_1, R_2, R_3) = (\{v_2, v_3, v_4\}, \{v_5\}, \{v_1\})$ achieves the values $(\alpha, \alpha(Z-1)/2Z)$ while $(S_1, S_2, S_3) = (\{v_2, v_3, v_5\}, \{v_4\}, \{v_1\})$ achieves the values $(\alpha(Z-1)/2Z, \alpha)$. Since $(Z-1)/2Z < 1/2$, the algorithm returns $(T_1, T_2, T_3) = (\{v_1, v_2, v_3\}, \{v_5\}, \{v_4\})$ and $W(T_1, T_2, T_3) = L(T_1, T_2, T_3) = \alpha(Z+1)/2Z$. When Z grows, (T_1, T_2, T_3) tends to be $(\alpha/2, \alpha/2)$-approximate.

Now suppose that $k > 3$, one can enrich the instance of Fig. 1 with isolated nodes and show that the analysis is tight.

6. Concluding remarks

Since we considered a bi-criteria MAX-CUT problem and provided approximation algorithms, the question whether it is possible to get similar results with more than two criteria arises. Unfortunately, the example given in Fig. 3 shows that, following the simultaneous approach, it is not possible to build a deterministic approximation algorithm with a performance guarantee when three (and more) criteria are considered. However, building a randomized one is still possible since the algorithm which consists in building a cut (S_1, S_2) by putting equiprobably a vertex $v \in V$ to either S_1 or S_2 remains a $\frac{1}{2}$-expected approximation algorithm for any multi-criteria version of the weighted MAX-CUT problem. Nevertheless, the existence of a polynomial time randomized algorithm that has a performance ratio strictly better than

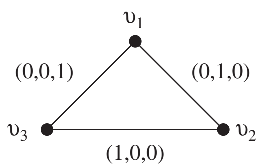

Fig. 3. The optimum value on each criterion is 1 while any feasible cut achieves 0 on at least one coordinate. Thus, no approximation factor can be guaranteed for a deterministic approximation algorithm.

1/2 for three and more criteria is open. As it was done in Proposition 1, one can build an instance which, for any number of criteria, shows that any randomized algorithm is at most $\frac{2}{3}$-expected approximate. As an example, one can consider the instance given in Fig. 3 and return the cut $(\{v_1, v_2, v_3\}, \{\})$ with a probability 0 while all the others are returned with a probability $\frac{1}{3}$.

Note that approximation results for the multi-criteria weighted MAX-CUT problem could be found if another approach is considered. Indeed, if we restrict ourselves to feasible solutions then rarely a solution will dominate all the others (i.e. will be better than the others on every criterion) but a set of solutions which dominate all the others always exists. This set of solutions is called the *Pareto curve* and Papadimitriou and Yannakakis [13] proved that an approximation with performance guarantee of this curve (an ε-*approximate Pareto curve*) always exists.

The algorithms proposed in this article achieve the same ratios for both criteria. Indeed, **Bi-Approx** is a $(\alpha/2, \alpha/2)$-approximation algorithm while **Ran Bi-Approx** is a $((\sqrt{5}-1)/2\alpha, (\sqrt{5}-1)/2\alpha)$-expected approximation algorithm. As a consequence, it would be interesting to obtain results with different ratios.

Acknowledgment

We thank Martin Skutella for giving us the example of Fig. 1.

References

[1] E. Angel, E. Bampis, A.V. Fishkin, A note on scheduling to meet two min-sum objectives, Proceedings of 9th International Workshop on Project Management and Scheduling (PMS'2004), 2004, pp. 143–146.
[2] J. Aslam, A. Rasala, C. Stein, N. Young, Improved bicriteria existence theorems for scheduling, Proceedings of 10th Annual ACM-SIAM Symposium on Discrete Algorithms, 1999, pp. 846–847.
[3] F. Baille, E. Bampis, C. Laforest, A note on bicriteria schedules with optimal approximations ratios, Parallel Process. Lett. 14 (2004) 315–323.
[4] E. Bampis, A. Kononov, Bicriteria approximation algorithms for scheduling problems with communications delays, J. Scheduling 8 (4) (2005) 281–294.
[5] F. Barahona, M. Grötschel, M. Jünger, G. Reinelt, An application of combinatorial optimization to statistical physics and circuit layout design, Oper. Res. 36 (1998) 493–513.
[6] M. Bruglieri, F. Maffioli, M. Ehrgott, Cardinality constrained minimum cut problems: complexity and algorithms, Discrete Appl. Math. 137 (2004) 311–341.
[7] M. Ehrgott, Multicriteria Optimization, Lecture Notes in Economics and Mathematical Systems N 491, Springer, Berlin, 2000.
[8] M.X. Goemans, D.P. Williamson, .879-approximation algorithms for MAX CUT and MAX 2SAT, in: Proceedings of STOC, 1994, pp. 422–431.
[9] M.X. Goemans, D.P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, J. ACM 42 (6) (1995) 1115–1145.
[10] J. Hromkovič, Algorithmics for Hard Problems: Introduction to Combinatorial Optimization, Randomization, Approximation, and Heuristics, Springer, Berlin, 2001.
[11] G. Jäger, A. Srivastav, Improved approximation algorithms for maximum graph partitioning problems, in: Proceedings of FSTTCS, 2004, pp. 348–359.
[12] R.M. Karp, Reducibility among combinatorial problems, in: R. Miller, J. Thatcher (Eds.), Complexity of Computer Computations, Plenum Press, NY, 1972, pp. 85–103.
[13] C.H. Papadimitriou, M. Yannakakis, On the approximability of trade-offs and optimal access of web sources, in: Proceedings of the 41th Annual IEEE Symposium on Foundations of Computer Science, 2000, pp. 86–92.
[14] A. Rasala, C. Stein, E. Torng, P. Uthaisombut, Existence theorems, lower bounds and algorithms for scheduling to meet two objectives, in: Proceedings of the 13th Annual ACM–SIAM Symposium on Discrete Algorithms, 2002, pp. 723–731.
[15] C. Stein, J. Wein, On the existence of schedules that are near-optimal for both makespan and total weighted completion time, Oper. Res. Lett. 21 (1997) 115–122.