Multiplying Pessimistic Estimators: Deterministic Approximation of Max TSP and Maximum Triangle Packing

Anke van Zuylen*

Institute for Theoretical Computer Science, 1-208 FIT Building, Tsinghua University,
Beijing 100084, China
anke@tsinghua.edu.cn

Abstract. We give a generalization of the method of pessimistic estimators [14], in which we compose estimators by multiplying them. We give conditions on the pessimistic estimators of two expectations, under which the product of the pessimistic estimators is a pessimistic estimator of the product of the two expectations. This approach can be useful when derandomizing algorithms for which one needs to bound a certain probability, which can be expressed as an intersection of multiple events; using our method, one can define pessimistic estimators for the probabilities of the individual events, and then multiply them to obtain a pessimistic estimator for the probability of the intersection of the events. We apply this method to give derandomizations of all known approximation algorithms for the maximum traveling salesman problem and the maximum triangle packing problem: we define simple pessimistic estimators based on the analysis of known randomized algorithms and show that we can multiply them to obtain pessimistic estimators for the expected weight of the solution. This gives deterministic algorithms with better approximation guarantees than what was previously known.

Keywords: derandomization, approximation algorithms, pessimistic estimators, maximum traveling salesman problem, maximum triangle packing.

1 Introduction

In this paper, we consider two NP-hard maximization problems: the maximum traveling salesman problem and the maximum triangle packing problem. For these two problems, the best known approximation algorithms are randomized algorithms: the solution they output has an *expected* weight that is at least a certain factor of the optimum. Practically, one may prefer deterministic approximation algorithms to randomized approximation algorithms, because the

* This work was supported in part by the National Natural Science Foundation of China Grant 60553001, and the National Basic Research Program of China Grant 2007CB807900, 2007CB807901.

M.T. Thai and S. Sahni (Eds.): COCOON 2010, LNCS 6196, pp. 60–69, 2010.

approximation guarantee of a randomized algorithm only says something about the expected quality of the solution; it does not tell us anything about the variance in the quality. Theoretically, it is a major open question whether there exist problems that we can solve in randomized polynomial time, for which there is no deterministic polynomial time algorithm. The algorithms for the problems we consider and their analysis get quite complicated, so complicated, even, that two papers on maximum triangle packing contained subtle errors, see [11,2]. A direct derandomization of these algorithms remained an open problem. In fact, Chen et al. [5] claim in the conference version of their paper on the maximum triangle packing problem that their "randomized algorithm is too sophisticated to derandomize". We show that this is not the case: our derandomizations are quite simple, and are a direct consequence of the analysis of the randomized algorithms, a simple lemma and the right definitions.

Two powerful tools in the derandomization of randomized algorithms are the method of conditional expectation [8] and the method of pessimistic estimators, introduced by Raghavan [14]. We explain the idea of the method of conditional expectation and pessimistic estimators informally; we give a formal definition of pessimistic estimators in Section 2. Suppose we have a randomized algorithm for which we can prove that the expected outcome has a certain property, say, it will be a tour of length at least L. The method of conditional expectation does the following: it iterates over the random decisions by the algorithm, and for each such decision, it evaluates the expected length of the tour conditioned on each of the possible outcomes of the random decision. By the definition of conditional expectation, at least one of these outcomes has expected length at least L; we fix this outcome and continue to the next random decision. Since we maintain the invariant that the conditional expected length is at least L, we will finish with a deterministic solution of length at least L. It is often difficult to exactly compute the conditional expectations one needs to achieve this. A pessimistic estimator is a lower bound on the conditional expectation that is efficiently computable and can be used instead of the true conditional expectation.

In the algorithms we consider in this paper, it would be sufficient for obtaining a deterministic approximation algorithm to define pessimistic estimators for the probability that an edge occurs in the solution returned by the algorithm. Since the algorithms are quite involved, this seems non-trivial. However, the analysis of the randomized algorithm does succeed in bounding this probability, by defining several events for each edge, where the probability of the edge occurring in the algorithm's solution is exactly the probability of the intersection of these events. With the right definitions of events, defining pessimistic estimators for the probabilities of these individual events turns out to be relatively straightforward. We therefore give conditions under which we can multiply pessimistic estimators to obtain a pessimistic estimator for the expectation of the product of the variables they estimate (or, in our case, the probability of the intersection of the events). This allows us to leverage the analysis of the randomized algorithms for the two problems we consider and gives rise to quite simple derandomizations. In the case of the algorithms for the maximum triangle packing problem [10,5], we also simplify a part of the analysis that is used for both the original randomized algorithm and our derandomization, thus making the analysis more concise.

The strength of our method is that it makes it easier to see that an algorithm can be derandomized using the method of pessimistic estimators, by "breaking up" the pessimistic estimator for a "complicated" probability, into a product of pessimistic estimators for "simpler" probabilities. This is useful if, as in the applications given in this paper, the analysis of the randomized algorithm also bounds the relevant probability by considering it as the probability of an intersection of simpler events, or put differently, the randomized analysis bounds the relevant probability by (repeated) conditioning.

1.1 Results and Related Work

It is straightforward to show, using linearity of expectation, that the sum of pessimistic estimators that estimate the expectation of two random variables, say Y and Z respectively, is a pessimistic estimator of the expectation of $Y+Z$. The product of these pessimistic estimators is not necessarily a pessimistic estimator of $\mathbb{E}\big[YZ\big]$, and we are not aware of any research into conditions under which this relation does hold, except for a lemma by Widgerson and Xiao [16]. In their work, a pessimistic estimator is an upper bound (whereas for our applications, we need estimators that are lower bounds on the conditional expectation), and they show that if we have pessimistic estimators for the probability of events σ and τ respectively, then the sum of the estimators is a pessimistic estimator for the probability of $\sigma \cap \tau$. Because we need estimators that are lower bounds, this result does not hold for our setting. The conditions we propose for our setting are quite straightforward and by no means exhaustive: the strength of these conditions is that they are simple to check for the two applications we consider. In fact, our framework makes it easier to prove the guarantee for the randomized algorithms for the maximum triangle packing problem as well.

Using our approach, we show how to obtain deterministic approximation algorithms that have the same performance guarantee as the best known randomized approximation algorithms for the maximum traveling salesman problem and the maximum triangle packing problem. In the maximum traveling salesman problem (max TSP) we are given an undirected weighted graph, and want to find a tour of maximum weight that visits all the vertices exactly once. We note that we do not assume that the weights satisfy the triangle inequality; if we do assume this, better approximation guarantees are known [12]. Max TSP was shown to be APX-hard by Barvinok et al. [1]. Hassin and Rubinstein [9] give a randomized approximation algorithm for max TSP with expected approximation guarantee $(\frac{25}{33}-\varepsilon)$ for any constant $\varepsilon > 0$. Chen, Okamoto and Wang [3] propose a derandomization, by first modifying the randomized algorithm and then using the method of conditional expectation, thus achieving a slightly worse guarantee of $(\frac{61}{81}-\varepsilon)$. In addition, Chen and Wang [6] propose an improvement of the Hassin and Rubinstein algorithm, resulting in a randomized algorithm with guarantee $(\frac{251}{331}-\varepsilon)$.

In the maximum triangle packing problem, we are again given an undirected weighted graph, and we want to find a maximum weight set of disjoint 3-cycles. The maximum triangle packing problem cannot be approximated to within 0.9929 unless $P = NP$ as was shown by Chleík and Chlebíková [7]. Hassin and Rubinstein [10] show a randomized approximation algorithm with expected approximation guarantee $(\frac{43}{83} - \varepsilon)$ for any constant $\varepsilon > 0$. Chen, Tanahashi and Wang [5] improve this algorithm to give a randomized algorithm with improved guarantee of $(0.5257 - \varepsilon)$. A recent result of Tanahashi and Chen [4], which has not yet appeared, shows an improved approximation ratio for the related maximum 2-edge packing problem. They show how to derandomize their new algorithm using the method of pessimistic estimators. The derandomization is quite involved and many parts of the proof requires a tedious case analysis. Their method also implies a deterministic algorithm with guarantee $(0.518 - \varepsilon)$ for the maximum triangle packing problem. Using our method, we can give a simpler argument to show their result.

2 Pessimistic Estimators

We use the following definition of a pessimistic estimator.

Definition 1 (Pessimistic Estimator). *Given random variables $X_1, \ldots, X_r$, and a function $Y = f(X_1, \ldots, X_r)$, a pessimistic estimator ϕ with guarantee y for $\mathbb{E}\big[Y\big]$ is a set of functions $\phi^{(0)}, \ldots, \phi^{(r)}$ where $\phi^{(\ell)}$ has domain $\mathcal{X}_\ell := \{(x_1, \ldots, x_\ell) : \mathbb{P}\big[X_1 = x_1, \ldots, X_\ell = x_\ell\big] > 0\}$ for $\ell = 1, 2, \ldots, r$ such that $\phi^{(0)} \geq y$ and:*

(i) $\mathbb{E}\big[Y | X_1 = x_1, \ldots, X_r = x_r\big] \geq \phi^{(r)}(x_1, \ldots, x_r)$ *for any* $(x_1, \ldots, x_r) \in \mathcal{X}_r$*;*

(ii) $\phi^{(\ell)}(x_1, \ldots, x_\ell)$ *is computable in deterministic polynomial time for any* $\ell \leq r$ *and* $(x_1, \ldots, x_\ell) \in \mathcal{X}_\ell$*;*

(iii) $\mathbb{E}\big[\phi^{(\ell+1)}(x_1, \ldots, x_\ell, X_{\ell+1})\big] \geq \phi^{(\ell)}(x_1, \ldots, x_\ell)$ *for any* $\ell \leq r$ *and* $(x_1, \ldots, x_\ell) \in \mathcal{X}_\ell$*;*

(iv) we can efficiently enumerate all $x_{\ell+1}$ *such that* $(x_1, \ldots, x_{\ell+1}) \in \mathcal{X}_{\ell+1}$ *for any* $\ell < r$ *and* $(x_1, \ldots, x_\ell) \in \mathcal{X}_\ell$ *.*

In this paper, we are concerned with using pessimistic estimators to derandomize approximation algorithms (for maximization problems). The random variables $X_1, \ldots, X_r$ represent the outcomes of the random decisions made by the algorithm, and Y is the objective value of the solution returned by the algorithm.

Lemma 2. *Given random variables $X_1, \ldots, X_r$, a function $Y = f(X_1, \ldots, X_r)$, and a pessimistic estimator ϕ for $\mathbb{E}\big[Y\big]$ with guarantee y, we can find $(x_1, \ldots, x_r)$ such that*

$$\mathbb{E}\big[Y | X_1 = x_1, \ldots, X_r = x_r\big] \geq y.$$

Proof. By conditions (ii)-(iv) in Definition 1, we can efficiently find x_1 such that $\phi^{(1)}(x_1) \geq \phi^{(0)} \geq y$, and by repeatedly applying these conditions, we find $(x_1, \ldots, x_r)$ such that $\phi^{(r)}(x_1, \ldots, x_r) \geq y$. Then, by condition (i) in Definition 1, it follows that $\mathbb{E}\big[Y | X_1 = x_1, \ldots, X_r = x_r\big] \geq y$. □

We make a few remarks about Definition 1. It is sometimes required that condition (i) holds for all $\ell = 0, \ldots, r$. However, the condition we state is sufficient. Second, note that we do not require that the support of the random variables is of polynomial size. This fact may be important, see for example Section 3. Also note that we can combine (ii), (iii) and (iv) into a weaker condition, namely that for any $\ell < r$ and $(x_1, \ldots, x_\ell) \in \mathcal{X}_\ell$ we can efficiently find $x_{\ell+1}$ such that $\phi^{(\ell+1)}(x_1, \ldots, x_\ell, x_{\ell+1}) \geq \phi^{(\ell)}(x_1, \ldots, x_\ell)$. Our stricter definition will make it easier to define under which conditions we can combine pessimistic estimators.

The following lemma is easy to prove.

Lemma 3. *If Y, Z are both functions of $X_1, \ldots, X_r$ and ϕ is a pessimistic estimator for $\mathbb{E}\big[Y\big]$ with guarantee y and θ is a pessimistic estimator for $\mathbb{E}\big[Z\big]$ with guarantee z, then $\phi + \theta$ is a pessimistic estimator for $\mathbb{E}\big[Y + Z\big]$ with guarantee $y + z$. Also, for any input parameter $w \in \mathbb{R}^{\geq 0}$, $w\phi$ is a pessimistic estimator of $\mathbb{E}\big[wY\big]$ with guarantee wy.*

In our applications, the input is an undirected graph $G = (V, E)$ with edge weights $w_e \geq 0$ for $e \in E$. Let Y_e be an indicator variable that is one if edge e is present in the solution returned by the algorithm, and suppose we have pessimistic estimators ϕ_e with guarantee y_e for $\mathbb{E}\big[Y_e\big] = \mathbb{P}\big[Y_e = 1\big]$ for all $e \in E$. By the previous lemma, $\sum_{e \in E} w_e \phi_e$ is a pessimistic estimator with guarantee $\sum_{e \in E} w_e y_e$ for $\mathbb{E}\big[\sum_{e \in E} w_e Y_E\big]$, the expected weight of the solution returned by the algorithm. Hence such pessimistic estimators are enough to obtain a deterministic solution with weight $\sum_{e \in E} w_e y_e$.

Rather than finding a pessimistic estimator for $\mathbb{E}\big[Y_e\big]$ directly, we will define pessimistic estimators for (simpler) events such that $\{Y_e = 1\}$ is the intersection of these events. We now give two sufficient conditions on pessimistic estimators so that we can multiply them and obtain a pessimistic estimator of the expectation of the product of the underlying random variables. We note that our conditions are quite specific and that the fact that we can multiply the estimators easily follows from them. The strength of the conditions is in the fact that they are quite easy to check for the pessimistic estimators that we consider.

Definition 4 (Product of Pessimistic Estimators). *Given random variables $X_1, \ldots, X_r$, and two pessimistic estimators ϕ, θ for the expectation of $Y = f(X_1, \ldots, X_r)$, $Z = g(X_1, \ldots, X_r)$ respectively, we define the product of ϕ and θ, denoted by $\phi \cdot \theta$, as*

$$\phi \cdot \theta^{(\ell)}(x_1, \ldots, x_\ell) = \phi^{(\ell)}(x_1, \ldots, x_\ell) \cdot \theta^{(\ell)}(x_1, \ldots, x_\ell) \text{ for } \ell = 1, \ldots, r.$$

Definition 5 (Uncorrelated Pessimistic Estimators). *Given random variables $X_1, \ldots, X_r$, and two pessimistic estimators ϕ, θ for the expectation of functions $Y = f(X_1, \ldots, X_r)$, $Z = g(X_1, \ldots, X_r)$ respectively, we say ϕ and θ are uncorrelated, if for any $\ell < r$ and $(x_1, \ldots, x_\ell) \in \mathcal{X}_\ell$, the random variables $\phi^{(\ell+1)}(x_1, \ldots, x_\ell, X_{\ell+1})$ and $\theta^{(\ell+1)}(x_1, \ldots, x_\ell, X_{\ell+1})$ are uncorrelated.*

Definition 6 (Conditionally Nondecreasing Pessimistic Estimators). *Given random variables $X_1, \ldots, X_r$, and two pessimistic estimators ϕ, θ for the* *expectation of functions* $Y = f(X_1, \ldots, X_r)$*,* $Z = g(X_1, \ldots, X_r)$ *respectively, we say that* ϕ is nondecreasing conditioned on θ *if* θ *only takes on non-negative values and*

$$\mathbb{E}\big[\phi^{(\ell+1)}(x_1, \ldots, x_\ell, X_{\ell+1}) \;\big|\; \theta^{(\ell+1)}(x_1, \ldots, x_\ell, X_{\ell+1}) = c\big] \geq \phi^{(\ell)}(x_1, \ldots, x_\ell)$$

for any $\ell < r$ *and* $(x_1, \ldots, x_\ell) \in \mathcal{X}_\ell$ *and* $c > 0$ *such that* $\mathbb{P}\big[\theta^{(\ell+1)}(x_1, \ldots, x_\ell, X_{\ell+1}) = c\big] > 0$*.*

The proof of the following two lemmas is straightforward, and is omitted due to space constraints.

Lemma 7. *Given two uncorrelated pessimistic estimators as in Definition 5 which take on only non-negative values, their product is a pessimistic estimator of* $\mathbb{E}\big[YZ\big]$ *with guarantee* $\phi^{(0)}\theta^{(0)}$*.*

Lemma 8. *Given two pessimistic estimators as in Definition 6, their product is a pessimistic estimator of* $\mathbb{E}\big[YZ\big]$ *with guarantee* $\phi^{(0)}\theta^{(0)}$*.*

Before we sketch how to use Definition 1 and Lemmas 2, 3, 7 and 8 to obtain deterministic algorithms for the max TSP and maximum triangle packing problems, we make the following remark.

Remark 9 (Random subsets instead of random variables). In the algorithms we consider, the random decisions take the form of random subsets, say $S_1, \ldots, S_r$, of the edge set E. We could define random variables $X_1, \ldots, X_r$ which take on n-digit binary numbers to represent these decisions, but to bypass this extra step, we allow $X_1, \ldots, X_r$ in the definition of pessimistic estimators to represent either random variables or random subsets of a fixed ground set.

3 Maximum Traveling Salesman Problem

In the maximum traveling salesman problem (max TSP), we are given a complete undirected graph $G = (V, E)$ with edge weights $w(e) \geq 0$ for $e \in E$. Our goal is to compute a Hamiltonian circuit (tour) of maximum edge weight. For a subset of edges E', we denote by $w(E') = \sum_{e \in E'} w(e)$. We let OPT denote the value of the optimal solution. A *cycle cover* (also known as a 2-factor) of G is a subgraph in which each vertex has degree 2. A maximum cycle cover is a cycle cover of maximum weight and can be computed in polynomial time (see, for instance, Chapter 10 of [13]). A *subtour* on G is a subgraph that can be extended to a tour by adding edges, i.e., each vertex has degree at most 2 and there are no cycles on a strict subset of V.

Both the algorithm of Hassin and Rubinstein [9] and the algorithm by Chen and Wang [6] start with an idea due to Serdyukov [15]: Compute a maximum weight cycle cover $\mathcal{C}$ and a maximum weight matching W of G. Note that the weight of the edges in $\mathcal{C}$ is at least OPT and the weight of W is at least $\frac{1}{2}OPT$. Now, move a subset of the edges from $\mathcal{C}$ to the matching W so that we get two subtours. Extend the two subtours to tours; at least one of the tours has weight at least $\frac{3}{4}OPT$.

The algorithms of Hassin and Rubinstein [9] and Chen and Wang [6] use the same idea, but are more complicated, especially the algorithm in [6]. Due to space constraints, we will only sketch the algorithm by Hassin and Rubinstein [9] here. The algorithm constructs three tours instead of two; the first tour is deterministic and we refer the reader to [9] for its construction. The second and third tours are constructed by a randomized algorithm which is inspired by Serdyukov's algorithm [15]: a subset of the edges from $\mathcal{C}$ is moved from $\mathcal{C}$ to W in such a way that we obtain two subtours. However, to deal with the case when a large part of the optimal tour consists of edges with endpoints in different cycles in $\mathcal{C}$, a third set of edges M is computed, which is a maximum weight matching on edges that connect nodes in different cycles in $\mathcal{C}$. In the last step of the algorithm, the algorithm adds as many edges from M as possible to the remaining edges of $\mathcal{C}$ while maintaining that the graph is a subtour. Finally, the two constructed subtours are extended to tours.

The part of the algorithm that is difficult to derandomize is the choice of the edges that are moved from $\mathcal{C}$ to W. These edges are chosen so that, in expectation, a significant fraction of the edges from M can be added in the last step. We will call a vertex *free* if one of the two edges adjacent to it in $\mathcal{C}$ is chosen to be moved to W. Note that if we add all edges of M for which both endpoints are free to the remaining edges in $\mathcal{C}$, then we may create new cycles. By the definition of M, such cycles must each contain at least two edges from M. Hence we know that we can add at least half the (weight of the) edges of M for which both endpoints are free without creating cycles. We therefore try to choose edges to move from $\mathcal{C}$ to W in order to maximize the weight of the edges in M for which both endpoints are free.

We first give the following lemma, which states the resulting approximation guarantee if the weight of the edges in M for which both endpoints are free is at least $\frac{1}{4}w(M)$. The proof of this lemma is implied by the proof of Theorem 5 in Hassin and Rubinstein [9].

Lemma 10 ([9]). *Given a weighted undirected graph $G = (V, E)$, let W a maximum weight matching on G and $\mathcal{C} = \{C_1, \ldots, C_r\}$ be a maximum weight cycle cover. Let M be a maximum weight matching on edges that connect nodes in different cycles in $\mathcal{C}$. If we can find nonempty subsets $X_1, \ldots, X_r$, where $X_i \subseteq C_i$ such that*

- *$(V, W \cup \bigcup_{i=1}^{r} X_i)$ is a subtour;*
- *the weight of the edges from M for which both endpoints are "free" (adjacent to some edge in $\bigcup_{i=1}^{r} X_i$) is at least $\frac{1}{4}w(M)$;*

then there exists a $(\frac{25}{33} - \varepsilon)$-approximation algorithm for max TSP.

Hassin and Rubinstein [9] succeed in finding random sets $X_1, \ldots, X_r$ such that the second condition holds in expectation: The algorithm considers the cycles in $\mathcal{C}$ one by one, and when considering cycle C_i in $\mathcal{C}$, they give a (deterministic) procedure (see the proof of Lemma 1 in [9]) for computing two non-empty candidate sets of edges in C_i, each of which may be added to $(V, W \cup \bigcup_{\ell=1}^{i-1} X_\ell)$ so that the resulting graph is a subtour. The candidate sets are such that each vertex in C_i is adjacent to at least one of the two candidate sets. Now, X_i is equal to one of these two candidate sets, each with probability 1/2. The two candidate sets from C_i depend on the edges $X_1, \ldots, X_{i-1}$ which were chosen to be moved from $C_1, \ldots, C_{i-1}$, and it is hence not clear if one can efficiently determine the probability distribution of the set of edges from C_i that will be moved to W. Fortunately, we do not need to know this distribution when using the appropriate pessimistic estimators.

Lemma 11. *There exists a deterministic* $(\frac{25}{33} - \varepsilon)$*-approximation algorithm for max TSP.*

Proof. For each vertex $v \in V$, we define the function Y_v which is 1 if v is adjacent to an edge in $\bigcup_{i=1}^{r} X_i$, and 0 otherwise. Let C_i be the cycle in $\mathcal{C}$ that contains v, and note that Y_v is a function of $X_1, \ldots, X_i$. We define a pessimistic estimator ϕ_v with guarantee 1/2 for the probability that Y_v is one. No matter which edges are in $X_1, \ldots, X_{i-1}$, the two candidate sets in C_i always have the property that each vertex in C_i is adjacent to at least one of the two candidate sets. We can therefore let $\phi_v(x_1, \ldots, x_\ell) = \frac{1}{2}$ if $\ell < i$, and we define it to be 0 or 1 depending on whether v has degree one if we remove the edges corresponding to $X_i = x_i$ from C_i if $i \geq \ell$. Clearly, ϕ_v satisfies properties (i)-(iii) in Definition 1 with respect to $\mathbb{P}\big[Y_v = 1\big]$. For (iv), note that given $X_1 = x_1, \ldots, X_{i-1} = x_{i-1}$, there are only two possible sets x_{i+1} (however, it is not necessarily the case that the random variable X_i has polynomial size support if we do not condition on $X_1, \ldots, X_{i-1}$).

Now, note that for an edge $\{u, v\} \in M$, u and v are in different cycles of the cycle cover. Therefore, for any $\ell < r$, either $\phi_u^{(\ell+1)}(x_1, \ldots, x_\ell, X_{\ell+1})$ or $\phi_v^{(\ell+1)}(x_1, \ldots, x_\ell, X_{\ell+1})$ is non-stochastic. Hence ϕ_u and ϕ_v are uncorrelated pessimistic extimators, and by Lemma 7, $\phi_u \cdot \phi_v$ is a pessimistic estimator with guarantee $\frac{1}{4}$ for $\mathbb{P}\big[Y_u = 1, Y_v = 1\big]$ which is exactly the probability that both endpoints of $\{u, v\}$ are free. By Lemmas 2, 3, we can use these pessimistic estimators to find subsets $x_1, \ldots, x_r$ that satisfy the conditions in Lemma 10. □

4 Maximum Triangle Packing

In the maximum triangle packing problem, we are again given an undirected graph $G = (V, E)$ with weights $w(e) \geq 0$ for $e \in E$. The goal is to find a maximum weight triangle packing, i.e., a maximum weight set of disjoint 3-cycles. We assume without loss of generality that $|V| = 3n$: otherwise, we can try all possible ways of removing one or two vertices so that the remaining number of vertices is a multiple of 3.

The algorithms for the maximum triangle packing problem by Hassin and Rubinstein [10] and Chen et al. [5] are similar in spirit to the algorithms for the max TSP. Due to space constraints, we again only sketch the ideas for the algorithm in [10]. The algorithm computes three triangle packings, and the first two are computed by deterministic algorithms. The third packing is computed by computing a subtour of large weight; note that we can find a triangle packing of weight equal to at least $2/3$ times the weight of any subtour.

The subtour is constructed in a similar way as in the previous section. First, a cycle cover $\mathcal{C}$ is computed; each cycle in the cycle cover has at most $1 + 1/\varepsilon$ edges and the weight of the cycle cover is large (at least $(1 - \varepsilon)$ times the weight of the maximum cycle cover). Also, a matching M is computed on the edges for which the endpoints are in different cycles of $\mathcal{C}$. Now, we remove a non-empty subset of the edges from each cycle in the cycle cover, and subsequently add edges from M for which both endpoints are "free", i.e. have degree zero or one. Denote the graph we obtain by G'. Note that G' may contain cycles, but each cycle contains at least two edges from M. To get a subtour, we choose one edge from M in each cycle (uniformly at random) and delete it.

The algorithm of Hassin and Rubinstein [10] removes edges from the cycles in the cycle cover by a random procedure in such a way that

(1) the expected weight of the edges removed from cycle $C \in \mathcal{C}$ is $\frac{1}{3}w(C)$ if $|C| = 3$, and $\frac{1}{4}w(C)$ otherwise;
(2) the probability that v becomes free (i.e. at least one edge in $\mathcal{C}$ that is adjacent to v is deleted) is at least $\frac{1}{2}$;
(3) the probability that $e \in M$ is added to G' (i.e. both endpoints of e are free) is at least $\frac{1}{4}$;
(4) conditioned on the event that e is in G', the probability that e is in a cycle in G' containing at most two other edges from M is less than the probability that e is not in a cycle in G'.

Note that (4) ensures that the probability that e is deleted from G', conditioned on both of its endpoints being free, is at most $\frac{1}{4}$. Hence, (3) and (4) ensure that the expected weight of the edges from M in the subtour is at least $\frac{1}{4} \times \frac{3}{4}w(M)$.

In the full version of this paper, we show how to define pessimistic estimators for the probability that each edge in $\mathcal{C} \cup M$ is in the subtour. For an edge in $\mathcal{C}$, this is straightforward. For edge $e = \{u, v\}$ in M, this probability is the probability of the intersection of the events that both u and v are free, and the event that edge e is not deleted from a cycle. We therefore define separate pessimistic estimators for the probabilities of these events: an estimator ϕ_u and ϕ_v for the probability that u and v, respectively, are free (each with guarantee $\frac{1}{2}$), and an estimator ϕ_{q_e} for the probability that an edge e is in the subtour, conditioned on the event that it was added to G' (with guarantee $\frac{3}{4}$). As in the previous section, the estimators ϕ_u, ϕ_v are uncorrelated for $e = \{u, v\} \in M$. Therefore, their product is a pessimistic estimator for the probability that e is added to G'. The estimator ϕ_{q_e}, that we define, turns out to be correlated with estimator $\phi_u \cdot \phi_v$. However, it is not hard to show that ϕ_{q_e} is nondecreasing conditioned on $\phi_u \cdot \phi_v$. Hence by Lemma 8, the product of $\phi_u \cdot \phi_v$ and ϕ_{q_e} is a pessimistic estimator for the probability that e is in the subtour. Hence we can find a deterministic algorithm that finds a subtour of weight at least as large as the expected weight of the tour by Hassin and Rubinstein's algorithm.

In the full version of this paper, we also apply our method to the randomized algorithms of Chen and Wang [6] and Chen et al. [5]. Finally, we give a simple proof of fact (4) listed above from the analysis of [10], which does not involve checking many cases as in the original proof.

Acknowledgements. The author thanks Bodo Manthey for pointing out that no direct derandomizations were known for the algorithms considered here.

References

1. Barvinok, A., Johnson, D.S., Woeginger, G.J., Woodroofe, R.: The maximum traveling salesman problem under polyhedral norms. In: Bixby, R.E., Boyd, E.A., Ríos-Mercado, R.Z. (eds.) IPCO 1998. LNCS, vol. 1412, pp. 195–201. Springer, Heidelberg (1998)
2. Chen, Z.-Z.: Personal Communication (2009)
3. Chen, Z.-Z., Okamoto, Y., Wang, L.: Improved deterministic approximation algorithms for Max TSP. Inform. Process. Lett. 95(2), 333–342 (2005)
4. Chen, Z.-Z., Tanahashi, R.: A deterministic approximation algorithm for maximum 2-path packing. To appear in IEICE Transaction
5. Chen, Z.-Z., Tanahashi, R., Wang, L.: An improved randomized approximation algorithm for maximum triangle packing. Discrete Appl. Math. 157(7), 1640–1646 (2009); Preliminary version appeared in AAIM '08
6. Chen, Z.-Z., Wang, L.: An improved randomized approximation algorithm for Max TSP. J. Comb. Optim. 9(4), 401–432 (2005)
7. Chlebík, M., Chlebíková, J.: Approximation hardness for small occurrence instances of NP-hard problems. In: Petreschi, R., Persiano, G., Silvestri, R. (eds.) CIAC 2003. LNCS, vol. 2653, pp. 152–164. Springer, Heidelberg (2003)
8. Erdõs, P., Spencer, J.: Probabilistic methods in combinatorics. Academic Press, New York (1974); Probability and Mathematical Statistics, vol. 17
9. Hassin, R., Rubinstein, S.: Better approximations for max TSP. Inform. Process. Lett. 75(4), 181–186 (2000)
10. Hassin, R., Rubinstein, S.: An approximation algorithm for maximum triangle packing. Discrete Appl. Math. 154(6), 971–979 (2006); Preliminary version appeared in ESA 2004
11. Hassin, R., Rubinstein, S.: Erratum to: An approximation algorithm for maximum triangle packing. Discrete Appl. Math. 154(6), 971–979 (2006); Discrete Appl. Math., 154(18), 2620 (2006)
12. Kowalik, L., Mucha, M.: Deterministic 7/8-approximation for the metric maximum tsp. Theoretical Computer Science 410, 5000–5009 (2009); Preliminary version appeard in APPROX-RANDOM '08
13. Lovász, L., Plummer, M.D.: Matching theory. North-Holland Mathematics Studies, vol. 121. North-Holland Publishing Co, Amsterdam (1986); Annals of Discrete Mathematics, 29
14. Raghavan, P.: Probabilistic construction of deterministic algorithms: approximating packing integer programs. J. Comput. System Sci. 37(2), 130–143 (1988)
15. Serdyukov, A.I.: An algorithm with an estimate for the travelling salesman problem of the maximum. Upravlyaemye Sistemy 89(25), 80–86 (1984)
16. Wigderson, A., Xiao, D.: Derandomizing the Ahlswede-Winter matrix-valued Chernoff bound using pessimistic estimators, and applications. Theory of Computing 4(1), 53–76 (2008)