Introduction to Domination Analysis

Gregory Gutin and Anders Yeo
Department of Computer Science
Royal Holloway, University of London
Egham, Surrey TW20 0EX, UK
gutin(anders)@cs.rhul.ac.uk

1 Introduction

In the recently published book on the Traveling Salesman Problem, half of Chapter 6 [18] is devoted to domination analysis (DA) of heuristics for the Traveling Salesman Problem. Another chapter [16] is a detailed overview of the whole area of DA. Both chapters are of considerable length. The purpose of this paper is to give a short introduction to results and applications of DA. While we do not prove any significant new results, we provide proofs to a few extensions and improvements of previously known theorems. Some open problems are also raised.

In Section 2 we provide motivation for studying DA. Basic terminology and notation are given in Section 3. The minimum multiprocessor scheduling problem is considered in some detail in Section 4. Section 5 overviews several results on other combinatorial optimization problems. Sections 6 and 7 are devoted to results on DA for the greedy algorithm and upper bounds on the maximum domination number for polynomial time traveling salesman heuristics.

2 Motivation

Research on combinatorial optimization (CO) heuristics has produced a large variety of heuristics especially for well-known CO problems. Thus, we should develop ways of selecting the best ones among them. In most of the literature, heuristics are compared in computational experiments. While experimental analysis is of definite importance, it cannot cover all possible families of instances of the CO problem at hand and, in particular, it usually does not cover the hardest instances.

The theoretical performance of CO heuristics is normally studied by the means of approximation analysis (AA) [3]. Usually, upper or lower bounds for the worst case performance ratio are obtained.

Introduced in [9], DA provides an alternative and a complement to AA. In DA, we are usually interested in the domination number or domination ratio of the heuristic solution. The *domination number (ratio)* of a heuristic H for a combinatorial optimization problem P is the maximum number (fraction) of all solutions that are not better than the solution found by H for any instance of P of size n.

In many cases, domination analysis is very useful. For example, the greedy algorithm has domination number 1 for many CO problems, see Section 6. In other words, the greedy algorithm, in the worst case, produces the unique worst possible solution. This is reflected in one of the latest computational experiments with the greedy algorithm, see, e.g., [19], where it was concluded that the greedy algorithm 'might be said to self-destruct'.

The **Asymmetric Traveling Salesman Problem (ATSP)** is the problem of computing a minimum weight tour (Hamilton cycle) in a weighted complete digraph on n vertices. The **Symmetric TSP (STSP)** is the same problem, but on a complete undirected graph. When a certain fact holds for both ATSP and STSP, we will simply speak of **TSP**. Sometimes, the maximizing version of TSP is of interest, which we denote by **max TSP**.

APX is the class of CO problems that admit polynomial time approximation algorithms with a constant performance ratio [3]. It is well known that while max TSP belongs to APX, TSP does not. This is at odds with the simple fact that a 'good' approximation algorithm for max TSP can be easily transformed into an algorithm for TSP. Thus, it seems that both max TSP and TSP should be in the same class of CO problems. The above asymmetry was already viewed as a drawback of performance ratio already in the 1970's, see, e.g., [6, 20, 23]. Notice that from the DA point view max TSP and TSP are equivalent problems.

Zemel [23] was the first to characterize measures of quality of approximate solutions (of binary integer programming problems) that satisfy a few basic and natural properties: the measure becomes smaller for better solutions, it equals 0 for optimal solutions and it is the same for corresponding solutions of equivalent instances. While the performance ratio and even the relative error (see [3]) do not satisfy the last property, the parameter $1 - r$, where r is the domination ratio, does satisfy all of the properties.

Local Search (LS) is one of the most successful approaches in the design of CO heuristics. Recently, several researchers carried out computational and theoretical investigation of LS with Very Large Scale Neighborhoods (see, e.g., [1, 7, 18]). The hypothesis 'justifying' this approach says that the larger the neighborhood the better solution is expected to be found [1]. However, some computational experiments do not support this hypothesis. This means that some other parameters are responsible for the relative power of a neighborhood. Computational experiments and theoretical results (see, e.g., [18, 22]) on TSP indicate that one such parameter may well be the domination ratio of the corresponding LS.

Sometimes, Approximation Analysis cannot be naturally used. Indeed, a large class of CO problems are multicriteria problems [8], which have several objective functions. (For example, consider STSP in which edges are assigned both time and cost, and one is required to minimize both time and cost.) We say that one solution s' of a multicriteria problems dominates another one s'' if the values of all objective functions at s' are not worse than those at s'' or the value of at least one objective function at s' is better than the value of the same objective function at s''. This definition allows us to naturally introduce the domination ratio (number) for multicriteria optimization heuristics. In particular, an algorithm that always finds a Pareto solution is of domination ratio 1.

In our view, it is advantageous to have bounds for both performance ratio and domination ratio of a heuristic whenever it is possible. Roughly speaking this will enable us to see a 2D picture rather than a 1D picture.

3 Basic Terminology

Let P be a CO problem, and let $\mathcal{H}$ be a heuristic for P. The **domination number** $\mathrm{domn}(\mathcal{H}, n)$ is the maximum integer $m(n)$ such that the solution $x(I)$ obtained by $\mathcal{H}$ for *any* instance I of P of size n is not worse than at least $m(n)$ feasible solutions of I (including $x(I)$). The **domination ratio** $\mathrm{domr}(\mathcal{H}, n)$ is the maximal $q(n)$ such that the solution $x(I)$ obtained by H for *any* instance I of P of size n is not worse than at least the fraction $q(n)$ of the feasible solutions of I. Clearly, domination ratio belongs to the interval $(0, 1]$ and exact algorithms are of domination ratio 1.

We illustrate the above definitions using the ATSP. The fact that the nearest neighbor algorithm (NN) for ATSP is of domination number 1 (first proved in [17]) means that for every $n \geq 2$, there is an instance of ATSP on n vertices, for which NN finds the *unique* worst possible Hamilton cycle. Since the number of distinct Hamilton cycles in an n-vertex complete digraph is $(n-1)!$, we see that NN is of domination ratio $1/(n-1)!$. There are many ATSP algorithms of domination number at least $(n-2)!$ [18], i.e., in the worst case they guarantee that their Hamilton cycle is at least as good as $(n-2)!-1$ other Hamilton cycles.

An algorithm $\mathcal{A}$ for a CO problem P is $\mathcal{DOM}$-**good** if $\mathcal{A}$ is of polynomial time complexity and there exists a polynomial p in n such that the domination ratio of $\mathcal{A}$ is at least $1/p(n)$ for any size n of P. A CO problem P is $\mathcal{DOM}$-**easy** if it admits a $\mathcal{DOM}$-good algorithm and $\mathcal{P}$ is $\mathcal{DOM}$-**hard** if there is no $\mathcal{DOM}$-good algorithm for P. The above mentioned algorithms for ATSP are of domination ratio $\Omega(1/(n-1))$ and thus ATSP is $\mathcal{DOM}$-easy.

4 Minimum Multiprocessor Scheduling Problem

Let $p \geq 2$ be an integer and let S be a finite set. A p-**partition** of S is a p-tuple $(A_1, A_2, \ldots, A_p)$ of subsets of S such that $A_1 \cup A_2 \cup \ldots \cup A_p = S$ and $A_i \cap A_j = \emptyset$ for all $1 \leq i < j \leq p$.

Let N denote the set $\{1, 2, \ldots, n\}$ and let each $i \in N$ be assigned a positive integral weight $\sigma(i)$. For a subset A of N, $\sigma(A) = \sum_{i \in A} \sigma(i)$. The **minimum multiprocessor scheduling problem (MMSP)** [3] can be stated as follows. We are given a triple (N, σ, p), where p is an integer, $p \geq 2$. We are required to find a p-partition $\mathcal{C}$ of N that minimizes $\sigma(\mathcal{A}) = \max_{1 \leq i \leq p} \sigma(A_i)$ over all p-partitions $\mathcal{A} = (A_1, A_2, \ldots, A_p)$ of N. We assume that $p < n$ as the opposite case is trivial.

As an illustration, we first consider a special case of MMSP when $p = 2$, which we denote M2SP and consider the following greedy-type algorithm $\mathcal{G}$ for the M2SP: $\mathcal{G}$ sorts the weights such that $\sigma(\pi(1)) \geq \sigma(\pi(2)) \geq \cdots \geq \sigma(\pi(n))$, initiates $A_1 = \{\pi(1)\}$, $A_2 = \{\pi(2)\}$, and, for each $j \geq 3$, puts $\pi(j)$ into the set A_j of the current 2-partition with smallest $\sigma(A_j)$, $j = 1, 2$. It is easy to see that any 2-partition (A_1, A_2) produced by $\mathcal{G}$ satisfies $|\sigma(A_1) - \sigma(A_2)| \leq \sigma(\pi(1))$. Consider any 2-partition (B_1, B_2) of $N - \{\pi(1)\}$ with the weights $\{\sigma(1), \sigma(2), \ldots, \sigma(n)\} - \{\sigma(\pi(1))\}$. If we add $\sigma(\pi(1))$ to B_2 if $\sigma(B_1) \leq \sigma(B_2)$ and to B_1, otherwise, then we obtain a 2-partition (C_1, C_2) for the original problem with $|\sigma(C_1) - \sigma(C_2)| \geq \sigma(\pi(1))$. Since the number of 2-partitions (C_1, C_2) constructed as above is at least half of all 2-partitions, the domination ratio of $\mathcal{G}$ is at least $\frac{1}{2}$.

Now we turn to the general MMSP. Notice that the size s of an instance of MMSP is $\Theta(n + \sum_{i=1}^{n} \log \sigma(i))$. Consider the following approximation algorithm $\mathcal{H}$ for MMSP. If $s \geq p^n$, then we simply solve the problem optimally. If $s < p^n$, then sort the elements of the sequence $\sigma(1), \sigma(2), \ldots, \sigma(n)$. For simplicity of notation, assume that $\sigma(1) \geq \sigma(2) \geq \cdots \geq \sigma(n)$. Compute $r = \lceil \log n / \log p \rceil$ and solve MMSP for $(\{1, 2, \ldots, r\}, \sigma, p)$ to optimality.

Suppose we have obtained a p-partition $\mathcal{A}$ of $\{1, 2, \ldots, r\}$. Now for i from $r+1$ to n add i to the set A_j of the current p-partition $\mathcal{A}$ with smallest $\sigma(A_j)$.

The following theorem proved in [11] generalizes a similar result for M2SP shown in [2].

Theorem 4.1 *The algorithm $\mathcal{H}$ runs in time $O(s^2 \log s)$. We have*

$$\lim_{s \longrightarrow \infty} \mathrm{domr}(\mathcal{H}, s) = 1.$$

5 Some Other Problems

We mentioned above that there are many ATSP heuristic of domination ratio $\Omega(1/n)$. However, it is not known yet whether there exists a polynomial time ATSP heuristic of domination ratio $\Omega(1)$. Recently, Alon, Gutin and Krivelevich [2] proved that some polynomial time algorithms for the weighted Max Cut, weighted Max k-SAT and several of its generalizations are of domination ratio $\Omega(1)$. In particular, they showed the following:

Theorem 5.1 *There exists a linear time, deterministic approximation algorithm for the weighted Max Cut whose domination ratio exceeds* $1/40$.

While the unweighted (i.e., all weights are equal 1) max SAT is proved to be $\mathcal{DOM}$-easy in [12], it is an open problem whether the weighted max SAT is $\mathcal{DOM}$-easy or not. Another problem of unknown 'status' is the Quadratic Assignment Problem (QAP). In [15], a polynomial time QAP algorithm is considered; it is proved that the algorithm is of domination ratio $\Omega(1/n^2)$ when n is a prime power. The result seems to be non-extendable to every n using the approach of [15].

Using a very powerful non-approximability result on Max Clique (see [3]) the following theorem was proved in [12].

Theorem 5.2 *The Max Clique and Min Vertex Cover are $\mathcal{DOM}$-hard unless P=NP.*

DA has been used for several other problems not mentioned above; for example, in [21], two algorithms for the Frequency Assignment Problem are compared using DA. A similar investigation for two Generalized ATSP heuristics was carried out in [5].

6 Greedy Algorithm

We mentioned above that the nearest neighbor algorithm for ATSP is of domination number 1. The same result holds for the greedy algorithm for ATSP [17]. The last result is extended to a wide class of CO problems in [14]. It turns out that the domination number of the greedy algorithm for (even) the Assignment Problem is 1.

In this section we prove an extension of the main theorem in [14]. Although our proof is similar to one in [14], we provide it for illustrative reasons. In the end of this section, we briefly discuss results of a recent paper [4].

An *independence system* is a pair consisting of a finite set E and a family $\mathcal{F}$ of subsets (called *independent sets*) of E such that (I1) and (I2) are satisfied.

(I1) The empty set is in $\mathcal{F}$;

(I2) If $X \in \mathcal{F}$ and Y is a subset of X, then $Y \in \mathcal{F}$.

A maximal (with respect to inclusion) set of $\mathcal{F}$ is called a **base**. Clearly, an independence system on a set E can be defined by its bases. Notice that bases may be of different cardinality.

Many combinatorial optimization problems can be formulated as follows. We are given an independence system $(E, \mathcal{F})$ and a weight function w that assigns a real weight $w(e)$ to every element $e \in E$. The weight $w(S)$ of $S \in \mathcal{F}$ is defined as the sum of the weights of the elements of S. It is required to find a base $B \in \mathcal{F}$ of minimum weight. In this section, we will consider only such problems and call them the $(E, \mathcal{F})$-**optimization problems**.

If $S \in \mathcal{F}$, then let $I(S) = \{x : \ S \cup \{x\} \in \mathcal{F}\} - S$. The greedy algorithm (or, `greedy`, for short) constructs a base as follows: `greedy` starts from an empty set X, and at every step `greedy` takes the current set X and adds to it a minimum weight element $e \in I(X)$; `greedy` stops when a base is built.

Note that if we add (I3) below to (I1),(I2), then we obtain one of the definitions of a matroid:

(I3) If U and V are in $\mathcal{F}$ and $|U| > |V|$, then there exists $x \in U - V$ such that $V \cup \{x\} \in \mathcal{F}$.

It is well-known that domination number of `greedy` for every matroid $(E, \mathcal{F})$ is $|\mathcal{F}|$: `greedy` always finds an optimum for the $(E, \mathcal{F})$-optimization problem. Thus, it is somewhat surprising to have the following theorem:

Theorem 6.1 *Let $(E, \mathcal{F})$ be an independence system and $B' = \{x_1, \ldots, x_k\}$, $k \geq 2$, a base. Suppose that the following holds for every base $B \in \mathcal{F}$, $B \neq B'$,*

$$\sum_{j=0}^{k-1} |I(x_1, x_2, \ldots, x_j) \cap B| < k(k+1)/2. \tag{1}$$

Then the domination number of `greedy` *for the $(E, \mathcal{F})$-optimization problem equals 1.*

Proof: Let M be an integer larger than the maximal cardinality of a base in $(E, \mathcal{F})$. Let $w(x_i) = iM$ and let $w(x) = 1 + jM$ if $x \notin B'$, $x \in I(x_1, x_2, \ldots, x_{j-1})$ but $x \notin I(x_1, x_2, \ldots, x_j)$. Clearly, `greedy` constructs B' and $w(B') = Mk(k+1)/2$.

Let $B = \{y_1, y_2, \ldots, y_s\}$ be a base different from B'. By the choice of w made above, we have that $w(y_i) \in \{a_iM, a_iM + 1\}$ for some positive integer a_i.

Clearly

$$y_i \in I(x_1, x_2, \ldots, x_{a_i - 1}),$$

but $y_i \notin I(x_1, x_2, \ldots, x_{a_i})$. Hence, by (I2), y_i lies in $I(x_1, x_2, \ldots, x_j) \cap B$, provided $j \leq a_i - 1$. Thus, y_i is counted a_i times in $\sum_{j=0}^{k-1} |I(x_1, x_2, \ldots, x_j) \cap B|$. Hence,

$$\begin{aligned} w(B) &= \sum_{i=1}^{s} w(y_i) \leq s + M \sum_{j=0}^{k-1} |I(x_1, x_2, \ldots, x_j) \cap B| \\ &\leq s + M(k(k+1)/2 - 1) = s - M + w(B'), \end{aligned}$$

which is less than the weight of B' as $M > s$. Since $\mathcal{A}$ finds B', and B is arbitrary, we see that `greedy` finds the unique heaviest base. $\diamond$

In [14], it is noted that the strict inequality (1) cannot be relaxed to the non-strict one as it is satisfied by some matroids.

Recall that by the Assignment Problem (AP) we understand the problem of finding a lightest perfect matching in a weighted complete bipartite graph $K_{n,n}$. Using Theorem 6.1, one can prove the following:

Corollary 6.2 *[14] Every greedy-type algorithm $\mathcal{A}$ is of domination number 1 for ATSP, STSP and AP.*

Bang-Jensen, Gutin and Yeo [4] considered the $(E, \mathcal{F})$-optimization problems, in which every base is of the same cardinality and w assumes only a finite number of integral values. For such problems, the authors of [4] completely characterized all cases when `greedy` *may* construct the unique worst possible solution. Here the word may means that `greedy` may choose any element of E of the same weight.

7 Upper Bounds for TSP Domination Number

In [13], we obtained upper bounds for the cardinality of polynomial time searchable ATSP neighborhoods. It turns out that minor changes in the proofs of [13] lead us to stronger and more general results on the maximum possible domination number of polynomial time ATSP heuristics. This is the topic of this section.

It is realistic to assume that any ATSP algorithm spends at least one unit of time on every arc of the complete digraph K_n^* that it considers. We use this assumption in the rest of this section.

Theorem 7.1 *Let $\mathcal{A}$ be an ATSP heuristic of complexity $t(n)$. Then the domination number of $\mathcal{A}$ does not exceed* $\max_{1\le n'\le n}(t(n)/n')^{n'}$.

Proof: Let $D = (K_n^*, w)$ be an instance of ATSP and let H be the tour that $\mathcal{A}$ returns, when its input is D. Let $DOM(H)$ denotes all tours in D which are not lighter than H including H itself. We assume that D is the worst instance for $\mathcal{A}$, namely $\text{domn}(\mathcal{A}, n) = |DOM(H)|$. Since $\mathcal{A}$ is arbitrary, to prove this theorem, it suffices to show that $|DOM(H)| \le \max_{1\le n'\le n}(t(n)/n')^{n'}$.

Let E denote the set of arcs in D, which $\mathcal{A}$ actually examines; observe that $|E| \le t(n)$ by the assumption above. Let F be the set of arcs in H that are not examined by $\mathcal{A}$, and let G denote the set of arcs in $D - A(H)$ that are not examined by $\mathcal{A}$.

We first prove that every arc in F must belong to each tour of $DOM(H)$. Assume that there is a tour $H' \in DOM(H)$ that avoids an arc $a \in F$. If we assign to a a very large weight, H' becomes lighter than H, a contradiction.

Similarly, we prove that no arc in G can belong to a tour in $DOM(H)$. Assume that an $a \in G$ and a is in a tour $H' \in DOM(H)$. By making a very light, we can ensure that $w(H') < w(H)$, a contradiction.

Now let D' be the digraph obtained by contracting the arcs in F and deleting the arcs in G, and let n' be the number of vertices in D'. Note that every tour in $DOM(H)$ corresponds to a tour in D' and, thus, the number of tours in D' is an upper bound on $|DOM(H)|$. In a tour of D', there are at most $d^+(i)$ possibilities for the successor of a vertex i, where $d^+(i)$ is the out-degree of i in D'. Hence we obtain that

$$|DOM(H)| \le \prod_{i=1}^{n'} d^+(i) \le \left(\frac{1}{n'}\sum_{i=1}^{n'} d^+(i)\right)^{n'} \le \left(\frac{t(n)}{n'}\right)^{n'},$$

where we applied the arithmetic-geometric mean inequality. $\diamond$

Corollary 7.2 *Let $\mathcal{A}$ be an ATSP heuristic of complexity $t(n)$. Then the domination number of $\mathcal{A}$ does not exceed* $\max\{e^{t(n)/e}, (t(n)/n)^n\}$*, where e is the basis of natural logarithms.*

Proof: Let $U(n) = \max_{1\le n'\le n}(t(n)/n')^{n'}$. By differentiating $f(n') = (t(n)/n')^{n'}$ with respect to n' we can readily obtain that $f(n')$ increases for $1 \le n' \le t(n)/e$, and decreases for $t(n)/e \le n' \le n$. Thus, if $n \le t(n)/e$, then $f(n')$ increases for every value of $n' < n$ and $U(n) = f(n) = (t(n)/n)^n$. On the other hand, if $n \ge t(n)/e$ then the maximum of $f(n')$ is for $n' = t(n)/e$ and, hence, $U(n) = e^{t(n)/e}$. $\diamond$

The next assertion follows directly from the proof of Corollary 7.2.

Corollary 7.3 *Let $\mathcal{A}$ be an ATSP heuristic of complexity $t(n)$. For $t(n) \ge en$, the domination number of $\mathcal{A}$ does not exceed $(t(n)/n)^n$.*

Note that the restriction $t(n) \ge en$ is important since otherwise the bound of Corollary 7.3 can be invalid. Indeed, if $t(n)$ is a constant, then for n large enough the upper bound becomes smaller than 1, which is not correct since the domination number is always at least 1.

It is proved in [10] that there are $O(n)$-time ATSP algorithms of domination number $2^{\Theta(n)}$. It follows from the last corollary that this result cannot be improved.

We finish this section with a result that improves (and somewhat clarifies) Theorem 20 in [22]. Our proof is a modification of the proof of Theorem 20 in [22].

Theorem 7.4 *Unless P=NP, there is no polynomial time ATSP algorithm of domination number at least $(n-1)! - \lfloor n - n^{\alpha}\rfloor!$ for any constant $\alpha < 1$.*

Proof: Assume that there is a polynomial time algorithm $\mathcal{H}$ with domination number at least $(n-1)! - \lfloor n - n^{\alpha} \rfloor!$ for some constant $\alpha < 1$. Choose an integer $s > 1$ such that $\frac{1}{s} < \alpha$.

Consider a weighted complete digraph (K_n^*, w). We may assume that all weights are non-negative as otherwise we may add a large number to each weight. Choose a pair u, v of vertices in K_n^*. Add, to K_n^*, another complete digraph D on $n^s - n$ vertices, in which all weights are 0. Append all possible arcs between K_n^* and D such that the weights of all arcs coming into u and going out of v are 0 and the weights of all other arcs are M, where M is larger than n times the maximum weight in (K_n^*, w). We have obtained an instance $(K_{n^s}^*, w')$ of ATSP.

Apply $\mathcal{H}$ to $(K_{n^s}^*, w')$ (observe that $\mathcal{H}$ is polynomial for $(K_{n^s}^*, w')$). Notice that there are exactly $(n^s - n)!$ Hamilton cycles in $(K_{n^s}^*, w')$ of weight L, where L is the weight of a lightest Hamilton (u, v)-path in K_n^*. Each of the $(n^s - n)!$ Hamilton cycles is obviously optimal. Observe that the domination number of $\mathcal{H}$ is at least $(n^s - 1)! - \lfloor n^s - (n^s)^{\alpha} \rfloor!$. However, for sufficiently large n, we have

$$(n^s - 1)! - \lfloor n^s - (n^s)^{\alpha} \rfloor! \geq (n^s - 1)! - (n^s - n)! + 1$$

as $n^{s\alpha} \geq n + 1$ for n large enough. Thus, a Hamilton cycle produced by $\mathcal{H}$ is always among the optimal solutions (for n large enough). This means that we can obtain a lightest Hamilton (u, v)-path in K_n^* in polynomial time, which is impossible since the lightest Hamilton (u, v)-path problem is a well-known NP-hard problem. We have arrived at a contradiction. $\diamond$

Acknowledgement Part of this paper was presented at Eurocomb'03 in Prague. The research of both authors was partially supported by a Leverhulme Foundation grant.

References

[1] R.K. Ahuja, Ö. Ergun, J.B. Orlin and A.P. Punnen, A survey of very large-scale neighborhood search techniques. *Discrete Appl. Math.* **123** (2002) 75–102.

[2] N. Alon, G. Gutin and M. Krivelevich, Algorithms with large domination ratio. *J. Algorithms* **50** (2004) 118–131.

[3] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela and M. Protasi, *Complexity and Approximation*, Springer, Berlin, 1999.

[4] J. Bang-Jensen, G. Gutin and A. Yeo, When the greedy algorithm fails. Submitted.

[5] D. Ben-Arieh, G. Gutin, M. Penn, A. Yeo and A. Zverovitch, Transformations of Generalized ATSP into ATSP: experimental and theoretical study. *Oper. Res. Lett.* **31** (2003) 357–365.

[6] G. Cornuejols, M.L. Fisher and G.L. Nemhauser, Location of bank accounts to optimize float; an analytic study of exact and approximate algorithms. *Management Sci.* **23** (1977) 789–810.

[7] V.G. Deineko and G.J. Woeginger, A study of exponential neighbourhoods for the traveling salesman problem and the quadratic assignment problem, *Math. Prog. Ser. A* **87** (2000) 519–542.

[8] M. Ehrgott and X. Gandibleux. A survey and annotated bibliography of multicriteria combinatorial optimization. *OR Spektrum* **22** (2000) 425–460.

[9] F. Glover and A. Punnen, The traveling salesman problem: New solvable cases and linkages with the development of approximation algorithms, *J. Oper. Res. Soc.* **48** (1997) 502–510.

[10] G. Gutin, Exponential neighbourhood local search for the traveling salesman problem. *Computers and Oper. Res.* **26** (1999) 313–320.

[11] G. Gutin, T. Jensen and A. Yeo, Domination analysis for minimum multiprocessor scheduling. Submitted.

[12] G. Gutin, A. Vainshtein and A. Yeo, Domination Analysis of Combinatorial Optimization Problems. *Discrete Appl. Math.* **129** (2003) 513–520.

[13] G. Gutin and A. Yeo, Upper bounds on ATSP neighborhood size. *Discrete Appl. Math.* **129** (2003) 533–538.

[14] G. Gutin and A. Yeo, Anti-matroids. *Oper. Res. Letters* **30** (2002) 97–99.

[15] G. Gutin and A. Yeo, Polynomial approximation algorithms for the TSP and the QAP with a factorial domination number. *Discrete Appl. Math.* **119** (2002) 107–116.

[16] G. Gutin and A. Yeo, Domination Analysis of Combinatorial Optimization Algorithms and Problems. In *Graph Theory, Combinatorics and Algorithms: Interdisciplinary Applications* (M.C. Golumbic and I. Ben-Arroyo Hartman, eds.), in preparation.

[17] G. Gutin, A. Yeo and A. Zverovitch, Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP. *Discrete Appl. Math.* **117** (2002) 81–86.

[18] G. Gutin, A. Yeo and A. Zverovitch, Exponential Neighborhoods and Domination Analysis for the TSP. In *The Traveling Salesman Problem and its Variations* (G. Gutin and A.P. Punnen, eds.), Kluwer, Dordrecht, 2002.

[19] D.S. Johnson, G. Gutin, L. McGeoch, A. Yeo, X. Zhang, and A. Zverovitch, Experimental Analysis of Heuristics for ATSP. In *The Traveling Salesman Problem and its Variations* (G. Gutin and A. Punnen, eds.), Kluwer, Dordrecht, 2002.

[20] H. Kise, T. Ibaraki and H. Mine, Performance analysis of six approximation algorithms for the one-machine maximum lateness scheduling problem with ready times. *J. Oper. Res. Soc. Japan* **22** (1979) 205–223.

[21] A.E. Koller and S.D. Noble, Domination analysis of greedy heuristics for the frequency assignment problem. To appear in *Discrete Math.*

[22] A.P. Punnen, F. Margot and S.N. Kabadi, TSP heuristics: domination analysis and complexity. *Algorithmica* **35** (2003) 111-127.

[23] E. Zemel, Measuring the quality of approximate solutions to zero-one programming problems. *Math. Oper. Res.* **6** (1981) 319–332.