$$
\begin{aligned}
&= -\frac{6}{n+1}(L - L_{av} \\
&= -\frac{6}{n+1}f.
\end{aligned}
$$

evaluate the correction term for each triple in the left side of table 1, but for brevity of exposition we will evaluate this term only for a sample of each class (T_2 and T_4).

Consider first the case of the triple in T_2 $(i, i, i+1)$. The correct value for $d_{i,i,i+1}$ is

$$-l_{i-1,i} - l_{i+1,i+2} + l_{i-1,i+1} + l_{i,i+2}.$$

The correction term due to the triples on the right side of table 1 is $-l_{i,i+1} + l_{i,i+1}$ for $(k, i, i+1)$, and $2l_{i,i+1}$ for $(i, k, i+1)$. The value of $d_{i,j,k}$ for $j = i$, and $k = i+1$, is $-l_{i-1,i} - 2l_{i,i+1} - l_{i+1,i+2} + l_{i-1,i+1} + l_{i,i+2}$. Summing up all the correction terms, we obtain $-l_{i-1,i} - l_{i+1,i+2} + l_{i-1,i+1} + l_{i,i+2}$, which is exactly the term we need for $(i, i, i+1)$.

The second case we consider is $(i, i+1, i+2)$, which belongs to the class T_4. The correct value for $d_{i,i+1,i+2}$ is

$$-l_{i-1,i} - l_{i,i+1} - l_{i+2,i+3} + l_{i-1,i+1} + l_{i+2,i} + l_{i,i+3}.$$

The correction term due to the triples on the right side of table 1 is $-l_{i,i+1} + l_{i+1,i+2}$ for $(i, i+1, k)$ and $2l_{i+1,i+2}$ for $(k, i, i+1)$. The value of $d_{i,j,k}$, for $j = i+1$ and $k = i+2$, is

$$-l_{i-1,i} - l_{i,i+1} - 2l_{i+1,i+2} - l_{i+2,i+3} + l_{i-1,i+1} + l_{i,i+2} + l_{i,i+3}.$$

Summing up all the correction terms and comparing them with the correct value of $d_{i,j,k}$, we obtain the global correction term $l_{i,i+1} - l_{i+1,i+2}$.

In general, for any fixed values of the index i, we have that $l_{i,i+1} - l_{i+1,i+2} \neq 0$, but

$$\sum_{i=1}^{n} (l_{i,i+1} - l_{i+1,i+2}) = 0.$$

We have not examined in great detail all the entries in Table 1, but only two of them which correspond to triples in T_2 and T_5, respectively. All the other triples lead to the same result.

The above considerations allow us to prove the theorem. Summing up the formulas for $d_{i,j,k}$, over i, j, and k ,we obtain

$$\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} d_{i,j,k} = -6n^2 L + 6n(n-1)L_{av}.$$

The correction terms are all equal to zero, except the correction term due to the triples in T_5, which is $6nL$. Class T_1 is not considered because no new solution is given by triples in T_1. The cardinality of the neighborhood is thus equal to $n^3 - n$. In conclusion, we have

$$\begin{aligned} \nabla^2 f &= \frac{-6n^2 L + 6n(n-1)L_{av} + 6nL}{n^3 - n} \\ &= \frac{6Ln(n-1) + 6n(n-1)L_{av})}{n(n-1)(n-2)} \end{aligned}$$

Triples	*Correction*	
$(i, i+1, i+2)$	$(i, i+1, k)$	$(k, i, i+1)$
$(i, i-1, i-2)$	$(i, i-1, k)$	$(k, i, i-1)$
$(i, i-2, i-1)$	$(i, k, i-1)$	$(k, i, i+1)$
$(i, i+1, i-1)$	$(i, i+1, k)$	$(i, k, i-1)$
$(i, i+2, i+1)$	$(k, i, i-1)$	$(i, k, i+1)$
$(i, i+2, i+1)$	$(i, i-1, k)$	$(i, k, i+1)$
$(i, i, i+1)$	$(k, i, i+1)$	$(i, k, i+1)$
$(i, i, i-1)$	$(k, i, i-1)$	$(i, k, i-1)$
$(i, i+1, i)$	$(i, i+1, k)$	$(k, i, i-1)$
$(i, i-1, i)$	$(i, i-1, k)$	$(k, i, i+1)$
$(i, i-1, i-1)$	$(i, i-1, k)$	$(i, k, i-1)$
$(i, i+1, i+1)$	$(i, i+1, k)$	$(i, k, i+1)$

Table 1: Double Correction. The first column contains triples of indexes which belong to T_2 and T_4. The second column shows triples of indexes which belong to T_5 for all the values of k, but the forbidden ones. In these cases, the triples in the second column become equal to the ones in the first column.

and k, and one term involving only the index i. The latter term contains the two consecutive indexes $i+1$ and $i+2$, and the two terms which involve i and k have different sign. These properties are satisfied by each triple of indexes in T_5. Summing up $l_{i+1,i+2} + l_{i+1,k} - l_{i,k}$ over all the triples in T_5, we obtain

$$\sum_{i=1}^{n} \sum_{k=1}^{n} (l_{i+1,i+2} - l_{i,k} + l_{i+1,k}) = nL. \tag{17}$$

For each type of triple in T_5, we get equality 17, so that the correction term for T_5 is $6nL$.

Equality 17 is true if the index k assumes all the values from 1 t o n. By the definition of T_5 we know that k cannot assume some particular values . When k assumes these "forbidden" values, we operate a correction to triples which belong to other classes. If these triples belong to a class that already needs a correction term, i.e. T_2 and T_4, then we operate a *double* correction. This has to be analyzed carefully.

When k assumes one of the "forbidden" values, then the triple we obtain belongs to the class T_2 or T_4. A description of this fact can be found in Table 1.

In the left side of table 1 there are the triples of indexes which belong to T_2 and T_4. T_2 and T_4 already need a correction term without considering the correction term due to triples of indexes in T_5 for k "forbidden"; in the right side of table 1 we have the triples of indexes which belong to T_5 and that, for particular values of the index k, became equal to the correspondent triples on the left of table 1.

When we look for the correction term for T_2 and T_4 we must consider the correction term already found for the triples in T_5 where k assumes those particular values. We

- $T_1 = \{(i,i,i), 1 \le i \le n\}, \#T_1 = n$;
- $T_2 = \{(i,i,i+1),(i,i,i-1),(i,i+1,i),(i,i-1,i),(i,i-1,i-1),(i,i+1,i+1), 1 \le i \le n\}, \#T_2 = 6n$;
- $T_3 = \{(i,i,k),(i,k,i),(k,i,i), 1 \le i \le n, 1 \le k \le n, k \ne i-1, k \ne i, k \ne i+1\}, \#T_3 = 3n(n-3)$;
- $T_4 = \{(i,i+1,i+2),(i,i-1,i-2),(i,i-2,i-1),(i,i+1,i-1),(i,i+2,i+1),(i,i-1,i+1), 1 \le i \le n\}, \#T_4 = 6n$;
- Let a and b be the two indexes (other than k) which appear in the definition of T_5.

  $T_5 = \{(i,i+1,k),(i,k,i+1),(k,i,i+1),(i,i-1,k),(k,i,i-1),(i,k,i-1)$, $1 \le i \le n, 1 \le k \le n, \mid k-a \mid > 1$ and $\mid k-b \mid > 1\}$, $\#T_5 = 6n(n-4)$;
- $T_6 = \{(i,j,k), \mid i-j \mid > 1, \mid i-k \mid > 1, \mid j-k \mid > 1\}, \#T_6 = n(n-5)(n-4)$;

One can readily verify that $\{T_i\}_{1 \le i \le 6}$ is a partition of T.

The class T_1 does not need a correction term. In fact $d_{i,i,i} = 0$, so that $d_{i,i,i}$ is exactly the cost obtained when the three indexes are equal. The same arguments hold for T_3 and T_6.

Classes T_2, T_4, and T_5 need a correction term. This term has to be be added to $d_{i,j,k}$. The basic idea to find the correction term is the following. Summing up the quantity $l_{i,i+1}$, for $1 \le i \le n$, we obtain exactly the cost L of the pivot solution, while summing up the quantity $l_{i,k}$, for $1 \le i \le n$ and $1 \le k \le n$, and using the fact that $L_{av} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} l_{i,j}}{(n-1)}$, we obtain $(n-1)L_{av}$.

We now evaluate the correction that must be added to $d_{i,j,k}$ for the triples which belong to the class T_5. We describe the procedure to obtain the correction term only for the triple $(i,i+1,k)$ of indexes in T_5[4]. If we evaluate $d_{i,i+1,k}$, we obtain

$$\begin{aligned} &-l_{i-1,i} - l_{i,i+1} - l_{i,i+1} - l_{i+1,i+2} - l_{k-1,k} - l_{k,k+1} \\ &+l_{i-1,i+1} + l_{i,k} + l_{k,i+2} + l_{k-1,i} + l_{i,k+1}, \end{aligned}$$

which is wrong.

The correct one is

$$\begin{aligned} &-l_{i-1,i} - l_{i,i+1} - l_{i+1,i+2} - l_{k-1,k} - l_{k,k+1} \\ &+l_{i-1,i+1} + l_{i+1,k} + l_{k,i+2} + l_{k-1,i} + l_{i,k+1}. \end{aligned}$$

Therefore, the appropriate correction term is $l_{i+1,i+2} + l_{i+1,k} - l_{i,k}$. This term has a particular structure, namely it is formed by two terms involving the indexes i

[4]The same procedure can be used in the other cases.

[9] C. H. Papadimitiou, K. Steiglitz. Combinatorial Optimization: Algorithms and complexity, *Prentice-Hall, Englewood Cliffs, NJ,* 1982.

[10] C. H. Papadimitiou, K. Steiglitz. On the complexity of local search for the traveling salesman problem, *SIAM J. Comput.,* 6, 1977.

Appendix

In this section we prove Theorem 3.

Proof of Theorem 3. Let s be a solution $\ldots i \ldots j \ldots k \ldots$ for the TSP, and $\overline{s}$ a solution in $N(s)$ given by $\ldots j \ldots k \ldots i \ldots$; then we define $d_{i,j,k}$ as $c(s) - c(\overline{s})$.

Note that, when two of the three indexes are equal, the 2+3-new-change neighborhood is equal to the 2-new-change neighborhood. The difference $d_{i,j,k} = c(\overline{s}) - c(s)$ is given by:

$$\begin{aligned} d_{i,j,k} &= -l_{i-1,i} - l_{i,i+1} - l_{j-1,j} - l_{j,j+1} - l_{k-1,k} - l_{k,k+1} \\ &\quad +l_{i-1,j} + l_{j,i+1} + l_{j-1,k} + l_{k,j+1} + l_{k-1,i} + l_{i,k+1}. \end{aligned} \tag{14}$$

Formula 14 needs to be corrected for particular values of the three indexes. As an example, for $i = j - 1$, we have

$$\begin{aligned} d_{i,i+1,k} &= -l_{i-1,i} - l_{i,i+1} - l_{i,i+1} - l_{i+1,i+2} - l_{k-1,k} - l_{k,k+1} \\ &\quad +l_{i-1,i+1} + l_{i+1,i+1} + l_{i,k} + l_{k,i+2} + lk - 1, i + l_{i,k+1}. \end{aligned} \tag{15}$$

This value is different from the correct one, which is

$$\begin{aligned} &-l_{i-1,i} - l_{i,i+1} - l_{i+1,i+2} - l_{k-1,k} - l_{k,k+1} \\ &+l_{k+1,i} + l_{i-1,i+1} + l_{i,k-1} + l_{i+1,k} + l_{k,i+2}. \end{aligned} \tag{16}$$

Thus the correction term is $+l_{i,i+1} + l_{i+1,k} - l_{i,k}$.

Comparing 14 and 16, it turns out that $d_{i,j,k}$ is wrong for particular value of i, j and k. We now partition the set of triples of indexes in classes which express the need of specific correction terms. Let T be defined as

$$T = \{(i,j,k), 1 \leq i,j,k \leq n\}.$$

$\{T_i\}_{1 \leq i \leq k}$ be a partition of T. Our goal is to find a partition such that $d_{i,j,k}$, evaluated over triples of indexes in T_i, either takes the exact value for each triple in T_i or takes a wrong value for each triple in T_i. In other words, we want to detect particular triples of indexes for which $d_{i,j,k}$, as defined in 14, is not correct. We introduce the following partition

$$\mathcal{P}(T) = \{T_1, T_2, T_3, T_4, T_5, T_6\},$$

where

4 Conclusion and Open Problems

It is an open question whether the relation shown for specific notion of neighborhoods can be generalized. Specifically, it is not clear which is the relation between the cost of a solution for the TSP and the parameter ∇^2, when the k-change, or the k-new-change, or even the 2+..+k-new-change neighborhood are used.

To this regard, we report some experimental evidences.

- The linear correlation coefficient calculated between $\nabla^2 f$ and f using the k-new- change neighbor assumes a value exactly equal to -1. This fact shows that 1 is satisfied; unfortunately the coefficient k of this equation is unknown. We only know that k seems not to be a costant value but a function of n.

- The linear correlation coefficient calculated between $\nabla^2 f$ and f using both the k-change and the 2+...+k-new-change is not exactly -1 but it is very close to -1.

Acknowledgements. We wish to thank Lov Grover for providing us with an updated version of [3], and Marco Pellegrini for reading an early version of this paper.

References

[1] S. A. Cook. The complexity of theorem-proving procedures, *Proc. 3rd Annual ACM Symp. Theory of Computing*, 151–158, 1971.

[2] M. R. Garey, D. S. Johnson. A guide to the theory of NP-completeness, *Freeman, San Francisco,* 1979.

[3] Lov. K. Grover. Neighborood search and the TSP, *Manuscript, School of Electrical Engineering, Cornell University,* 1989. Revised version.

[4] Lov. K. Grover. Local search and the local structure of NP-complete problems, *Manuscript, School of Electrical Engineering, Cornell University,* 1989.

[5] D. S. Johnson, C. H. Papadimitriou, M. Yannakakis. How easy is local search ?, *J. Comp. Syst. Sci.,* 37(1):79–100, 1988.

[6] M. W. Krentel. The complexity of optimization problems, *J. Comp. Syst. Sci.,* 36(3):490–509, 1988.

[7] M. W. Krentel. On finding and verifying locally optimal solution, *Fourth annual IEEE, Structure in complexity theory,* 132–137, 1989.

[8] C. H. Papadimitiou, K. Steiglitz. Some examples of difficult traveling salesman problem, *Oper. Res.,* 26:434–443, 1978.

Theorem 3 *The TSP and 2+3-new-change neighborhood satisfy equation*

$$\nabla^2 f + \frac{6}{n+1} f = 0.$$

Proof See the Appendix. ∎

Theorem 4 *The TSP and 3-new-change neighborhood satisfy equation*

$$\nabla^2 f + \frac{6}{n} f = 0.$$

Proof We do not show directly that the TSP and 3-new-change satisfy 1, but we prove it using the fact that 2-new-change and 2+3-new-change neighborhoods satisfy 1.

The cardinality of 2+3-new-change neighborhood is $n^3 - n$. It contains all the triples of indexes, except those that do not individuate any neighbor solution, i.e. (i, i, i). In particular, 2+3-new-change contains all the triples of indexes that determine a swap of only two cities; thus the set of solutions that belong to the 2+3-new-change neighborhood contains the set of solutions which belong to the 2-new-change neighborhood. Note that for each pair (i, j) in 2-new-change neighborhood, we have more than one triple of indexes in the 2+3-new-change neighborhood that determines the same solution. In particular we have exactly 3 triples which give the same solution. We can associate to the pair (i, j) the triples (i, i, j), (i, j, i), and (j, i, i). We call A, B, and C the cardinality of the 2+3-new-change, the 2-new-change, and the 3-new-change, respectively. Note that this quantity is equal to $n(n-1)(n-2)$. We call ∇^2_A, ∇^2_B, and ∇^2_C, the difference operator over the 2+3-new-change, the 2-new-change, and the 3-new- change, respectively.

We have

$$A\nabla^2_A = B\nabla^2_B + C\nabla^2_C,$$

or

$$\nabla^2_A(n^3 - n) = 3\nabla^2_B n(n-1) + \nabla^2_C n(n-1)(n-2). \tag{13}$$

From 13 it follows that

$$\nabla^2_C = \frac{\nabla^2_A(n^3 - n) + 3\nabla^2_B n(n-1)}{n(n-1)(n-2)}.$$

Using theorems 2 and 3, we obtain

$$\begin{aligned}\nabla^2_C &= \frac{\frac{6}{n+1}(n^3 - n) - \frac{12}{n} n(n-1)}{n)n-1)(n-2)} \\ &= \frac{6n(n-1) - 12(n-1)}{n(n-1)(n-1)} \\ &= \frac{6}{n}.\end{aligned}$$

∎

In conclusion, the TSP with the 3-new-change neighborhood satisfies 1, for $k = 6$.

Proof Consider a solution s, called *pivot solution*. Let $s_{i,j}$ be the solution obtained from s taking off the arcs $(i, i+1)$ and $(j, j+1)$, and replacing them with the arcs (i, j) and $(i+1, j+1)$. Let $d_{i,j} = c(s) - c(s_{i,j})$. The quantity $d_{i,j}$ is defined as

$$d_{i,j} = -l_{i,i+1} - l_{j,j+1} + l_{i,j} + l_{i+1,j+1}. \tag{11}$$

This formula is correct for any value of i and j, except the case $i = j$. In this case a correction term must be added to $d_{i,j}$. This term can be calculated comparing the wrong and the correct formula for $d_{i,j}$. In fact, $d_{i,i}$ calculated by 11, is equal to $2l_{i,i+1}$, while $d_{i,i}$ would be equal to 0. It turns out that, the correction term which must be added to $d_{i,j}$, when $i = j$, is $-2l_{i,i+1}$. We also observe that

$$\sum_{i=1}^{n} d_{i,j} = \sum_{\substack{j=1 \\ j \neq i}}^{n} d_{i,j}.$$

Now we can say that

$$\nabla^2 f = \frac{\sum_{i=1}^{\mu} \delta_i}{\mu} = \frac{\sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} d_{i,j}}{n(n-1)} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} d_{i,j}}{n(n-1)},$$

and, substituting $d_{i,j}$ according to 11, we obtain

$$\begin{aligned} \nabla^2 f &= \frac{2 \sum_{i=1}^{n} \sum_{j=1}^{n} l_{i,j}}{n(n-1)} - \frac{2n}{n(n-1)} L + \frac{2}{n(n-1)} L \\ &= \frac{2}{n} L_{av} - \frac{2}{n} L = -\frac{2}{n}(L - L_{av}) \\ &= -\frac{2}{n} f. \end{aligned} \tag{12}$$

To obtain equality 12, we have used the fact that

$$L_{av} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} l_{i,j}}{(n-1)},$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} l_{i,i+1} = nL,$$

and

$$\sum_{i=1}^{n} 2l_{i,i+1} = 2L.$$

∎

Summarizing, we have shown that the TSP with the 2-change neighborhood satisfies equation 1, for $k = 2$

We give now a relation among $L(x)$, $LI(x)$ and L_{av}. We show that one and only one of the three following relations is true:

$$L(x) < LI(x) < L_{av}, \tag{8}$$
$$L(x) = LI(x) = L_{av}, \tag{9}$$
$$L(x) > LI(x) > L_{av}. \tag{10}$$

Equation 3 can be written as

$$LI = L + \beta(L_{av} - L), \qquad \beta = \frac{k}{n} > 0.$$

We examine the following three cases.

1. If $L_{av} = L$, then $LI = L$. Thus $L = LI = L_{av}$. (Only 8 is true)

2. If $L_{av} < L$, then $LI = L - \delta$ with $\delta > 0$, so that $LI < L$; it remains to show that $LI > L_{av}$, but we know that $LI = L(1-\beta) + \beta L_{av}$. Therefore, exploiting the fact that $L_{av} < L$, we have $LI > L_{av}(1-\beta) + L_{av}\beta = L_{av}$; in conclusion if $L_{av} < L$, then $LI < L$ and $LI > L_{av}$. (Only 9 is true).

3. If $L_{av} > L$, then $LI = L + \delta$, $\delta > 0$, so that $LI > L$; it remains to show that $LI < L_{av}$, but we know that $LI = L(1-\beta) + L_{av}\beta$. Therefore, since $L_{av} > L$, we have $LI < L_{av}(1-\beta) + \beta L_{av} = L_{av}$; in conclusion, if $L_{av} > L$, then $LI > L$ and $LI < L_{av}$.(Only 10 is true).

In the rest of this section we show that the TSP satisfies 1 with other notions of neighborhoods than those introduced in [3].

Theorem 1 *[4] The TSP with the 2-new-change neighborhood satisfies 1, with $k = 4$, i.e.*

$$\nabla^2 f + \frac{4}{n} f = 0,$$

where $f = L - L_{av}$. ∎

We show that the TSP satisfies equation 1 also with:

- *2-change* neighborhood.
- *2+3-new-change* neighborhood.
- *3-new-change* neighborhood.

Theorem 2 *The TSP and 2-change neighborhood satisfy equation*

$$\nabla^2 f + \frac{2}{n} f = 0.$$

We set $a = (1-\beta)$ and $b = \beta L_{av}$[3]. Substituting a and b in 3, we obtain

$$LI = aL(x) + b, \qquad a > 0, b > 0. \tag{4}$$

We now switch our attention to the TSP and we estimate the value of b. First of all, we give a formula for the TSP which expresses the average cost of the solutions. It can be shown that

$$L_{av} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} d(i,j)}{(n-1)}.$$

We denote with l_{av} the average distance between two cities, i.e.

$$l_{av} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} d(i,j)}{n^2}.$$

We have

$$b = \frac{kL_{av}}{n} \approx \frac{kn^2}{n(n-1)} l_{av} \approx kl_{av}.$$

When n is very large, then k_{lav} becomes a good approximation to b ; i.e.,

$$\forall \epsilon > 0 \ \exists \ \overline{n} \ s.t. \ \forall n > \overline{n}, \ \mid b - kl_{av} \mid < \epsilon.$$

Replacing b with kl_{av} in 4, we obtain

$$LI = aL + kl_{av}.$$

We denote by LI the average cost of the solutions which belong to the neighborhood of x, and with $\overline{LI}$ the average cost of the solutions which belong to the neighborhood of another solution $\overline{x}$ (in the same way we define the quantities L and $\overline{L}$). Our goal is to establish a relation between $(\overline{LI} - LI)$ and $(\overline{L} - L)$. To this extent we evaluate 3 for $\overline{x}$ and for x, respectively. We obtain

$$\overline{LI} = a\overline{L} + b, \tag{5}$$

and

$$LI = aL + b. \tag{6}$$

Subtracting the left and the right side of 6 from 5 (note that a and b independent of x and $\overline{x}$) we obtain

$$\overline{LI} - LI = a(\overline{L} - L). \tag{7}$$

In a local search procedure, we find a sequence of solutions $s_1, s_2, \ldots, s_k, \ldots$, with decreasing costs $L_1, L_2, \ldots, L_k, \ldots$. Equality 7 gives us the possibility to evaluate the correspondent values $LI_1, LI_2, \ldots, LI_k, \ldots$, with a unitary computational cost.

[3] a and b depend only on the specific instance

From the definition of the operator ∇^2 it follows that $\nabla^2(L - L_{av}) = \nabla^2(L)$. Then equation 1 becomes

$$\nabla^2(L) + \frac{k}{n}L = \frac{k}{n}L_{av},$$

or also

$$\alpha\nabla^2(L) + L = L_{av}, \tag{2}$$

where $\alpha = \frac{n}{k}$.

We observe that the average of costs over all the feasible solutions, which depends only on the problem instance, is obtained by adding the cost of a generic solution s and a quantity $\alpha\nabla^2 L$ depending on the costs of the solutions which belong to the neighborhood of s.

Minimizing the cost function of a combinatorial optimization problem which satisfies equation 1 is equivalent to maximizing the quantity $\alpha\nabla^2 L$.

A possible application of equation 2 is the following. A lower bound on the local minimum L_m (and then on the global minimmum G_m) can be found if the quantity $\alpha\nabla^2 L$ is less than a costant value h. In this case no local optimum L_m with cost less than $L_{av} - h$ exists.

Equation 2 gives a relation between the cost of a generic solution x, the average cost of all the solutions, and the parameter $\nabla^2 L$, which depends only on the costs of the solutions that belong to the neighborhood of x.

We now determine a relation between L_{av}, L, and a new parameter $LI(x)$. We denote by $LI(x)$ the average of the cost computed over the solutions that belong to the neighborhood of x. Let μ be the cardinality of $N(x)$. The difference equation 2 can be rewritten as follows

$$\frac{\sum_{i=1}^{\mu} \delta_i}{\mu} + \frac{k}{n}f = 0.$$

We indicate by f_i the difference between $L(x_i)$ and $L(x)$, where x_i is the i^{th} solution in the neighborhood of x. Replacing δ_i with $f_i - f$, we obtain

$$\frac{\sum_{i=1}^{\mu}(f_i - f)}{\mu} + \frac{k}{n}f = \frac{\sum_{i=1}^{\mu}(L(x_i) - L(x))}{\mu} + \frac{k}{n}f =$$
$$\frac{\sum_{i=1}^{\mu} L(x_i)}{\mu} - L(x) + \frac{k}{n}(L(x) - L_{av}) = 0.$$

If $\beta = \frac{1}{\alpha} = \frac{k}{n}$, then we get

$$LI - L(x) + \beta(L(x) - L_{av}) = LI - L(x)(1 - \beta) - \beta L_{av} = 0,$$

i.e.

$$LI = (1 - \beta)L(x) + \beta L_{av}. \tag{3}$$

Grover has also proved the existence of problems and neighborhoods which satisfy equation 1. The TSP is one of these problems. In this paper, we use the following formulation of the TSP.

A TSP is a complete weighted undirected graph $G = (N, d)$, where N is a set of nodes and d is a distance function which maps pair of nodes into real numbers. We assume that d satisfies the following three properties.

1. $d(i,j) = d(j,i), \qquad \forall i,j \in N;$
2. $d(i,j) \geq 0, \qquad \forall i,j \in N;$
3. $d(i,j) + d(j,k) \geq d(i,k), \qquad \forall i,j,k \in N.$

The number $d(i,j)$ is the weight of the edge (i,j). If we consider the set of nodes N as the set of the first n natural numbers (n = problem size), then we say that a feasible solution s for the TSP is any permutation of these numbers. The cost $C(s)$ of a solution for the TSP is given by

$$C(s) = \sum_{j=1}^{n} d(i_j, i_{j+1}) + d(i_n, i_1),$$

where $s = (i_1, i_2, \ldots, i_n)$, and i_j is a permutation of $1, 2, \ldots, n$. We also use the following notions of neighborhood.

- *k-change.* Given a TSP A and a solution s for A, we call *k-change(s)* the set of solutions obtained from s by substituting h arcs in s with h arcs not in s, $h \leq k$.
- *k-new-change.* Given a solution s for the TSP, we obtain another solution $\overline{s}$ in the neighborhood of s removing *exactly* k cities from s, and replacing them changing their position in an arbitrary way.
- *2+3+..+k-new-change.* Given a solution s for the TSP, we obtain another solution $\overline{s}$ in the neighborhood of s removing *at most* k cities in s, and replacing them changing their position in an arbitrary way.

3 Main results

In this section we analyze the local structure of the problems which satisfy equation 1 and we prove that the TSP with certain notions of neighborhood satisfies 1. If we replace f with $(L - L_{av})$, then equation 1 becomes

$$\nabla^2 (L - L_{av}) + \frac{k}{n}(L - L_{av}) = 0.$$

that the TSP with certain notions of neighborhood satisfies the difference equation. In Section 4 we give the above described experimental evidences.

2 Difference equation

Grover has shown in [4] that several NP-complete problems satisfy a simple linear difference equation that is similar to the wave equation of mathematical physics. Before introducing such equation, we need some preliminary definitions. Let L be the cost function of a combinatorial optimization problem,[1] and L_{av} the average over all the costs of all the feasible solutions to this problem. We denote with $f(x)$ the value of $L(x) - L_{av}$. In [4] it is shown that the TSP, min-cut graph partitioning, graph coloring, and a version of SAT satisfy the difference equation

$$\nabla^2 f + \frac{k}{n} f = 0, \qquad k > 0, \tag{1}$$

where n is the problem size, and k is a costant depending on the problem. The symbol $\nabla^2 f$ is used to denote the average difference operator over a specified neighborhood. If the solution x has X *neighbors*[2], then

$$\nabla^2 f(x) = \frac{\sum_{i=1}^{X} \delta_i}{X},$$

where δ_i is defined as

$$\delta_i = f(x) - f(x_i),$$

$x_i \in N(x)$, and x is called "pivot solution" or "configuration".

The difference equation 1 gives a relation between the cost $f(x)$ of a solution x and $\nabla^2 f(x)$, i.e. the costs of the solutions belonging to the neighborhood of x. This "link" between $f(x)$ and $\nabla^2 f(x)$ gives us some information about the local structure of the problems that satisfy 1. This information is *local* because at each step of the local search procedure, we have some relation between the solution x and its neighbors, but we can not establish any relation between x and the previous or the next solutions examined in the local search procedure.

Grover has obtained the following results [3, 4].

1. Any local optimum, for problems which satisfy 1, has a cost which is smaller than the average cost over all solutions. This means that arbitrarily poor local optima do not exist.

2. An algorithm does exist which, starting from an arbitrarily poor configuration, reaches a solution with a cost at least as good as the average cost, using $O(n)$ iterations.

[1] $L(x)$ denotes the cost of x.

[2] Solutions belonging to the neighborhood

to finding local optima. But no such approach has been discovered Yet. Nevertheless, local search algorithms based on neighborhood structures are not polynomial time guaranteed. In practice, however, these methods converge to the local optima very quickly and usually provide solutions whose cost is close to the global optima.

The notion of reduction and equivalence among PLS problems have been defined. A PLS problem is said to be PLS-complete if all PLS problems can be reduced to it in polynomial time. It has been proved that complete problems for the PLS class do exist. PLS-complete problems are equivalent with respect to finding the local optima. This means that, if the local optima for any of these problems can be found in polynomial time, then the local optima for all PLS-complete problems could be found in polynomial time. This fact suggests that these problems should share some properties.

Grover in [3, 4] has proved that certain NP-complete problems, (i.e. TSP, with a specific notion of neighborhood), satisfy a linear difference equation. Such equation gives a linear relation between the cost of a solution x and the costs of the solutions which belong to the neighborhood of x. Grover also shows that the TSP with *2-new-change* neighborhood satisfies the above mentioned difference equation. In this paper we analyze the difference equation presented in [3, 4] and we show that if a problem satisfies it, then the following structural properties are true.

- The average of costs over all the feasible solutions depends on the cost of any solution x, and on the costs of the solutions which belong to the neighborhood of x.
- The average of costs over the solutions which belong to the neighborhood of any solution x is linearly related to the cost of x.
- Let LI, L_{av}, and $L(x)$ be the average of costs over the solutions which belong to to the neighborhood of any solution x, the average of costs over all the feasible solutions, and the cost of x, respectively. LI, L_{av}, and $L(x)$ satisfy one and only one of the three following relations.
    1. $L(x) < LI(x) < L_{av}$
    2. $L(x) = LI(x) = L_{av}$
    3. $L(x) > LI(x) > L_{av}$

We prove that the TSP with *2-change*, *2+3-new-change*, and *3-new-change* notions of neighborhood satisfies the difference equation introduced in [3, 4]. Finally, we give experimental evidences suggesting that the results obtained for neighborhoods having a certain cardinality could be generalized for neighborhoods of any cardinality.

The rest of this paper is organized as follows. In Section 2 we recall some results obtained in [3, 4]. In Section 3 we manipulate the difference equation introduced in [4] to show some structural properties of the problems which satisfy it. We also prove

1 Introduction

An important class of computational problems is that of *NP-complete* problems. No polynomial time algorithm is known to solve an NP-complete problem. An approach to deal with NP-complete problems is to use heuristic algorithms, which, instead of providing a global optimum, simply find a "good" solution. When an upper bound is known on the gap between this solution and the optimal one, heuristic algorithms are called *approximation* algorithms.

In this paper we focus our attention on a popular heuristic for solving optimization problems, i.e. *local search*. We show that, if a problem satisfies a simple difference equation, then the solutions obtained by local search heuristics have interesting structural properties. We also prove that the Traveling Salesman Problem (TSP) with certain notions of neighborhood (*2-change*, *2+3-new-change*, and *3-new-change*) satisfies the above mentioned difference equation. We give now some preliminary definitions.

An *istance of an optimization problem* is a pair (F, c), where F is the set of all the feasible solutions to the problem instance, and c is a cost function. A solution $x \in F$, for which $c(x) \leq c(y)\ \forall y \in F$, is called a globally optimal solution to (F, c) or, when no confusion can arise, simply an optimal solution.

An *optimization problem* is a set $\{(F_i, c), i \in I\}$, where I is a set of indexes and (F_i, c) is a problem instance. We say that an optimization problem is an NP-complete problem if its decisional version is an NP-complete problem.

Given a feasible solution $x \in F$, it is useful to define a set $N(x)$ of feasible solutions that are close in some sense to x. $N(x)$ is called "neighborhood" of x. Given an optimization problem with instances $(F_i, c)_{i \in I}$, a neighborhood is a function $N : F_k \rightarrow 2^{F_k}$, defined for each instance F_k.

Finding a globally optimal solution to an NP-complete problem is very hard; given a suitable notion of neighborhood N, it is often possible to find, in polynomial time, a local optimum, i.e. a solution x with a cost $c(x)$ lesser than the cost of the solutions in $N(x)$. The kind of heuristic technique we analyze here is local search. It can be expressed as follows.

1. Find an initial feasible solution s.

2. Choose a solution $\overline{s} \in N(s)$, such that $c(\overline{s}) \leq c(s)$.

3. Repeat step 2 until $c(\overline{s}) \leq c(t),\ \forall t \in N(\overline{s})$.

In [5] a class of problems, called PLS problems (*polynomial time local search*) is defined. This class contains those local search problems for which local optimality can be verified in polynomial time.

In [5] it is shown that finding locally optimal solutions is easier than finding globally optimal solutions, unless NP=co-NP. Thus, it seems unlikely that $PLS = NP_s$; on the other hand, if $PLS = P_s$, then there probably exists a general approache

Local Properties of Some NP-Complete Problems *

B. Codenotti† L. Margara‡

TR-92-021

April 1992

Abstract

It has been shown that certain NP-complete problems, i.e. TSP, min cut, and graph partitioning, with specific notions of neighborhood, satisfy a simple difference equation. In this paper, we extend these results by proving that TSP with *2-change*, *2+3-new-change*, and *3-new-change* notions of neighborhood satisfy such a difference equation, and we derive some properties of local search when performed with the above definitions of neighborhood.

*This work has been partly supported by the Italian National Research Council, under the "Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo", subproject 2 "Processori dedicati".

†International Computer Science Institute, Berkeley, CA 94704, and Istituto di Elaborazione dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa (Italy).

‡International Computer Science Institute, Berkeley, CA 94704, and Dipartimento di Informatica, Università di Pisa (Italy).