OPTIMAL CONSENSUS RANKING USING SLS: AN APPROACH AND AN APPLICATION

Sira Allende[1]*, Carlos Bouza* and Marja Li**
*Facultad de Matemática y Computación, Universidad de La Habana
**Sofware Development Division, Institute of Computing Training.

ABSTRACT
This paper presents a methodology for reaching a consensus vector when different variables are ranked in a set of individuals. It suggests using Stochastic Local Search implemented by a Metaheuristic. The data provided by the ranking of journals are analyzed and the behavior of the Metaheuristics studied. As a result the influence of the parameters involved in the Metaheuristics is established. Statistical methods are used for deriving the significance of the parameters in the optimal solutions generated and for establishing which set of parameters has the same distribution function of the computing time.

KEYWORDS: ranking, SLS, Tabu Search, Simulated Annealing, Anova, Kolmogorov-Smirnov

MSC: 90C27

RESUMEN
En este trabajo **se** presenta una metodología para obtener un vector de consenso cuando diferentes variables son rankeadas en un conjunto de individuos. Se sugiere usar Búsqueda Estocástica Local implementada por una Metaheurística. Los datos provenientes del rankeo de revistas son analizados y el comportamiento de las Metaheurísticas estudiado. Como resultado la influencia de los parámetros de las Metaheurísticas es establecida. Métodos estadísticos son usados para derivar la significación de los parámetros en las soluciones óptimas generadas y en establecer que conjunto de parámetros poseen la misma función de distribución función del tiempo de cómputo.

1. INTRODUCTION

A common decision problem is to select the ”best” element of a set of possible alternatives with respect to some comparison criteria. Even if only a finite number of possible alternatives are considered, the problem is not easy-computing , because the best alternative respect to one criterion, could be the worst respect to another one and all criteria should be considered for the selection. For instance, if a group of specialists discuss a clinical case. Consider for example the case of an old woman with arthritis-rheumatoid; their opinions are not necessarily coincident when symptoms and laboratory analysis are evaluated. Nevertheless, it is necessary to get a group consensus based on an agreement. Another troublesome discussion appears as this disease can be treated by using different medicaments and the physicians must select one of them considering all the characteristic of the medicament and the patient. Both situations could be represented by a discrete Multicriteria decision model: a finite set of alternatives and a finite set of criteria (experts or attributes of the alternatives). For each criterion k, a preference relation P_k, $k = 1, \ldots, K$ is defined. The goal is to combine the preference relations P_k, $k = 1, \ldots, K$ in order to obtain a compromise or consensus preference relation P.

Other examples appear in voting systems and in marketing enterpriser and military activities. Preference relations could define a pre-order on the set of alternatives or not. In the first case, it is possible to rank them. Frequently, it is possible to express preference for an alternative over another, but not the degree of the preference. In this case, the preference relation is called ordinal; otherwise, the preference is called cardinal. Consequently, the methods for devising consensus preferences are classified as ordinal or cardinal methods. Ordinal methods require less and easier available information than cardinal methods. It is is very

[1] sira@matcom.uh.cu

advantageous in medical applications, because the information is, in many cases, qualitative and subjective. If the preference relations P_k are not described by numerical data, it is easier to use an ordinal than a cardinal preference relation, additionally the errors risk is less.

Section 2 we discuss on ranking consensus reaching and present the usual procedures based on deriving an "estimation" of it through computing a central tendency measure.
Section 3 is concerned with the discussion on Stochastic Local Search and the use of Metaheuristics for deriving optimal solutions for NP-hard problems. Section 4 presents the study of the consensus ranking of journals on economics using different indexing measures. A statistical procedure, allowing establishing the influence in the changes of the parameters of the Metaheuristics, is developed.

The obtained results provide an insight on the role of the current indexing procedures and on the effectiveness of using optimization techniques instead of relying on a single statistical measure for evaluating the importance of the publications. This methodology should allow evaluating journals and papers not included in the leading indexing systems.

2. THE SEARCH OF A CONSENSUS

2.1 Some early approaches

The problem of obtaining a consensus in the ranking is very old. Borda in 1781 developed a consensus ranking for voting in the Convention in France. Different approaches have been suggested in voting which lead to a compromise or consensus ranking problem.

Different problems need of ranking and producing a consensus ranking. See for example Black (1958), Cook and Seiford (1978). Numerous approaches have been suggested in the literature for aggregating individual rankings for determining a consensus.

Borda-Kendall's method is perhaps the agreement method most widely used. Kendall (1962) studied the ordinal ranking problem as a problem of estimation and proposed to rank the alternatives according to the mean of the ranks assigned by the criteria. It was actually the method proposed by Borda in the 18th century for candidates election, hence the method has been named Borda-Kendall method. It doesn't satisfy the axiom of independence and allows the compensation by criteria. However, Borda Kendall method fixes a total preorder.

Condorcet, also in the 18th century, established a positional voting system based on the pairwise comparison of alternatives. Alternative a_r wins a pairwise majority vote election against alternative a_s if more criteria rank *i* after *j* than *j* after *i*. Condorcet's winner is the alternative that has a simple majority over every other candidate. This agreement function does not allow the compensation by criteria. It could be described by a $n \times n$ matrix $A=(a^*_{ij})$, where $a^*_{ij} = 1$ iff a_i wins a_j . If the Condorcet's winner exits, Borda-Kendall's method might not put it in the highest place.

Coppeland see Cook et al (2005), proposed to rank the alternatives according to

$$\sum_i a^*_{ij} - \sum_j a^*_{ij}$$

It is the number of victories of the alternative i minus the number of its defeats. Coppeland's agreement function defines a preorder, if Condorcet's agreement function defines a pre-order, both pre-orders coincide.

Other agreement functions inspired in voting systems are based also in pairwise comparison between alternatives. Some of them lead to Borda-Kendall rank-vector, for example the so called Rank-differences, For/again votes.

Different coefficients have been defined for expressing agreement or disagreement measures. Consensus preference relation could be devised either comparing the coefficients values or using methods of mathematical programming.

2.2. Agreement and disagreement measures

The number of criteria for which alternative a_i is preferred over alternative a_j describes a consensus notion. If the criteria differ in importance, their weights have to be taken in account.

Using these ideas some coefficients have been defined. The following definition allows considering classes of them:

Definition Let

$$I_k(a_i, a_j) = \begin{cases} 1 \; if \; a_i P I_k a_j \\ 0 \; otherwise \end{cases}$$

The Ordinal Agreement Coefficient with the preference of a_i over a_j is given by:

$$O_{Ag}(i,j) = \sum_k w_k I_k(a_i, a_j)$$

The Ordinal Disagreement Coefficient with the preference of a_i over a_j is given by:

$$O_{Dg}(i,j) = \sum_k w_k \left(1 - I_k(a_i, a_j)\right) \sum_k w_k I_k(a_i, a_j)^*$$

Given an utility function U_k (in particular, a rank vector) associated to criterion
k, $k = 1, \ldots, K$, the Concordance Coefficient with the preference with the preference of a_i over a_j is given by:

$$C \quad (i,j) = \sum_k w_k Max \quad \left(0, U_k(a_i) - U_k(a_j) \quad \right)$$

The Discordance Coefficient with the preference with the preference of a_i over a_j is given by:

$$D \quad (i,j) = \frac{Max_k Max\left(0, U_k(a_i) - U_k(a_j) \quad \right)}{Max_{k,i,j}\left(0, U_k(a_i) - U_k(a_j) \quad \right)}$$

Using the coefficients defined we are able to define the superiority of an index of the alternatives as
:

Superiority index of alternative i= $SI \quad (i) = \sum_j \; C(i,j)$

Then we can compare a decide which alternative is superior using the decision rule

$a_i \; overheads \; a_j \; iff \; O_{Ag}(i,j) = \vartheta_a \; and \; O_{Dg}(i,j) < \vartheta_d, \vartheta_h \; a \; fixed \; threshold, h = a, d.$

Overhead relations are used by the softwares ELECTRE and PROMOTHEE for fixing consensus preferences.

2.3. Mathematical Programming Model Structures in deriving a Ranking Consensus

For obtaining agreement there have been formulated also mathematical programming models. Some of them are based on alternatives comparison, other deal with the minimization of a distance defined on the rank-vector's set.

Another way of modeling consensus in the ranking is to consider that a couple of DM`s fix a matrix

$$A_{ij}(k) = \left(a_{ij}(k)\right), a_{ij}(k) = \begin{cases} 1 \; if \; i \; and \; j \; assign \; different \; rank \; to \; k \\ \frac{1}{2} \; if \; i \; and \; j \; assign \; the \; same \; rank \; to \; k \\ 0 \; otherwise \end{cases}$$

Bowman-Colantoni, see Cook et al (2005) modeled the consensus preference relation by defining

$$X = \left(x_{ij}\right), a_{ij}(k) = \begin{cases} 1 \; if \; a_i P a_j \\ \frac{1}{2} \; if \; a_i I a_j \\ 0 \; otherwise \end{cases}$$

This matrix describes a pre-order. we look for the maximization of the function:

$$\sum_{i,j} x_{ij} C \quad (i,j)$$

This model presents a limitation: two optimal solutions of the model might represent preference relations not equivalents.

Analogous models could be formulated using the ordinal disagreement, concordance or discordance coefficients.

Borda's method is based on computing the sum of the ranks $R_i(u)$, assigned to item $u \in U$ by DM $i \in I$.. The consensus rank of u is

$$R_{u(B)} = \sum_{i=1}^{|I|} n_u R_i(u)$$

Consider that an expert or decision maker (DM) analyzes a set U of items ranks every $u \in U$ and provides a vector $R = \left(r_1, \dots, r_{|U|}\right)$. If we have a set I of DM`s a collection $R_1 = \left(r_{11}, \dots, r_{1|U|}\right)$,…, $R_{|I|} = \left(r_{|I|1}, \dots, r_{|I||U|}\right)$ and distance $d(R_i,R_{i*})$ can be defined for evaluating the agreement between a pair of DM`s. If we look for a consensus ranking the problem is to obtain an arrangement of the ranks. In the case where ties are permitted, the set of allowable rank positions is given by $W = \{1, 1.5, 2, 2.5 \ldots\ldots (2/U/ - 1), /n/\}$. the value or worth of being ranked in position k is some (generally unknown) quantity W_k. in the case of no ties, the alternatives are ranked only at positions $\{1, 2, 3, \ldots, /n/\}$. A set of alternatives β would normally occupy rank positions $k_1, k_1, +1 \ldots k_2$, would, if tied, be usually ranked at the median or the mean of these $k_2 - k_1 + I$ positions.

Consensus among a set of ranks can be looked using a distance function $d(R,R^*))$. Hence if V is the set of the permutations of the possible ranks we deal with the optimization problem

$$Min_V \sum_{i \in I} \sum_{u \in U} d(R_i(u), R_{opt}(u))$$

The distance function could be defined in different ways consequently will be defined the decision variables. This is natural approach as the consensus is obtained by obtaining the rank vector which minimizes the distance. V is the set of possible rankings.

A large set of axioms can be used for evaluating the goodness of a distance function. Following Cook -Seiford (1978) are considered the axioms:

Axiom 1: $d(A, B) \geq 0$, with equality iff $A \equiv B$.
Axiom 2: $d(A, B) = d(B, A)$.
Axiom 3: $d(A, C) \leq d(A, B) + d(B, C)$ for any three rankings *A, B, C,* with equality holding iff ranking *B* is between *A* and *C*.
Axiom 4: $d(A, B) = d(A^*, B^*)$, where A^* and B^* result from *A* and *B* respectively by the same permutation of the alternatives in each case.
Axiom 5: If A^* and B^* result from *A* and *B* by listing the same $(n + 1)th$ alternative in last place, then $d(A^*, B^*) = d(A, B)$.
Axiom 6: The minimum positive distance is 1.

Formally a mathematical approach to the Borda-Kendall problem is to solve the L_2 optimization problem

$$Min_{\vec{R}\in V} \sum_{i\in I} \sum_{u\in U} (R_i(u), R_{opt}(u))^2$$

Its solution minimizes the squared error. Only if we are ranking without ties the optimal solution of this problem and Borda-Kendall rank-vector coincide

Le us consider the L_1 optimization problem. That is we deal with the so called City Block (Manhattan) distance $d(R_i,R_c)=/R_i-R_c/$ It is the only distance satisfying the 6 axioms. We seek for solving

$$Min_{\vec{R}\in V} \sum_{i\in I} \sum_{u\in U} \left|R_i(u), R_{opt}(u)\right|$$

Following the approach and notation of Cook et al. (1996) is worthy to fix that the pure linear ordering is the case where the larger the rank the better.

The general quasi linear case define two sets of variables. $\{w_k\}_{k=1}^{|U|}$ and $\{u_r\}$, with the latter being a set of $/U/(/U/-1)$, $\{u_{k_1}\}_{k_1=1}^{|U|-1}$, $\{u_{k_1,k_1+1,k_1+2}\}_{k_1=1}^{|U|-2},\ldots,\{u_{k_1}\}_{k_1=1}^{2}, u_{1,2,\ldots|U|}$, where $u_{k_1,k_1+1,\ldots,k_1+r}$ represents the value or utility associated with the rank position occupied by r +i alternatives which, if separately ranked, would be positioned at $k_1, k_1 + 1. , k_1 + r.$ Rather than assuming that the only valid rank positions in a weak ordering are the integer points and the midpoints of t as in the case in the conventional model

Define the set of constraints

C(CB)

$W_k - U_{k+1} \geq Z$
$U_{k\,k+1} - W_{k+1} \geq Z, k=1,\ldots., /U/-1.$
If $r + 2$ alternatives are tied for positions $\{k, k + 1, \ldots, k + r + 1\}$, then the value $u_{k,k+1,\ldots,k+r}$
$u_{k,k+1,\ldots,k+r} - u_{k,k+1,\ldots,k+r+1} \geq Z$
$u_{k,k+1,\ldots,k+r+1} - u_{k+1,\ldots,k+r+1} \geq Z$, $k=1,\ldots., /U/-r-1$, $r=1,\ldots,/U/-2$
$w/U/ \geq Z,$
$\sum_{k=1}^{|U|} w_k = \frac{|U|(|U|+1)}{2}$ □

We consider Z as a parameter, discrimination parameter, and the last constraint fixes that $0 < Z \leq 1/2$. The other constraints establish that we discriminate between the worth of consecutive rank positions by some positive amount.

Note that if two alternatives, for example, are tied for positions k and $k + 1$, then the value U_k $k < 1$ of being ranked at this 'tied position' must be between the value W_k of being ranked at position k and the value W_k+1 associated with position $k + 1$.

Cook et al (1996) proved that the optimal consensus ranking is independent of Z and hence this constraint is unimportant. We have the optimization problem

$$P(CB) = \left\{ Min_{\{V_{CB}\}} \sum_{i \in I} \sum_{u \in U} |V_{iu} - V_{cu}|;\ |C(CB)} \right\},$$

$$V_{CB} = set\ of\ the\ defined\ w_k.u_k\ as\ corresponding$$

Theorem 3.1. of Cook et al (1996) established that the values (*W* or *u)* associated with the different rank positions are restricted only to be a natural ordering.

The ranking is obtained as solution of an assignment problem. Mathematical programming models provide an elegant formulations of the consensus preference problem. However, as we deal with discrete variables and the dimensional limitation of the software for such problems could be a handicap for their application to real problems. Establishing a consensus preference, does not mean necessarily that a good response to the problem has been found. Depending of the problem's nature, it could be necessary to count with DM`s information.

In the next section we will consider the problem of computing a consensus ranking,

3. COMBINATORIAL OPTIMIZATION PROBLEMS (COP)

3.1. The nature of COP

Many optimization problems become unmanageable using combinatorial methods as the number of objects becomes large.

From a decision theory point of view we have a cost function and a finite set of solutions. Each solution can be obtained by a permutation, an arrangement or a partition of the objects. For each instance we have a set of feasible solutions $S=\{s\}$ and f assigns to each pair (instance, feasible solution) a value of the involved objective function f. A COP looks for optimizing *f*.

The term local search identifies Metaheuristics that solve NP- hard optimization problems. They are useful for finding an optimal solution of an optimization problem searching a set of candidate solutions, search space.

In the sequel an important role is played by the concept of neighborhood.

Definition (Neighborhood). A mapping $\mathcal{N}: S \mapsto 2^S$ determines the set of neighboring solution of $s \in S$ $(\mathcal{N}(s) \subseteq S \backslash s)$. □

A local search algorithm (LSA) moves from solution to solution in the search space by applying local changes A LSA starts from an initial candidate solution and then moves iteratively to a neighbor solution using some logic procedure. Usually the set of neighbors of a candidate solution has several candidates. A choice is made using only the information provided by the solutions in the neighborhood of a selected candidate. The search ends whether the best solution selected is not improved in a given number of steps or the fixed time for the search is consumed.

A mathematical description in terms of a target function $f(\underline{x})$, $\underline{x}$ a vector with real or discrete coordinates. When x has only discrete coordinates we can model the problem using a graph.

We are involved in the search of an optimum it is needed distinguishing between the global and the local optimum

Definition. A solution $s \in S$ is a local optimum with respect to a neighborhood $\mathcal{N}(s)$ iff there does no exist another $s^* \in S$ such that $f(s^*)$ is better solution than $f(s)$. It is a global optimum if this statement is true for any $s^* \in S$.□

A LSA relies on neighborhoods that generally are of small size and its performance will be improved if there is a path connecting the feasible solutions with the global optimum.

If a neighbor solution is selected by taking the one locally optimal solution the Metaheuristic is named hill climbing. It moves from the initial solution attempting to find a better solution by incrementally changing a single element of the solution. A solution is considered as better solution if the new solution improves the previous one. It is repeated until there are no more improvements. Hill climbing is adequate if is enough finding a local optimum but is possible that the global optimum is not attained. More complex schemes have been developed for attenuating this inadequacy.

Note in the figure that the first summit can be identified as a global maximum being false.

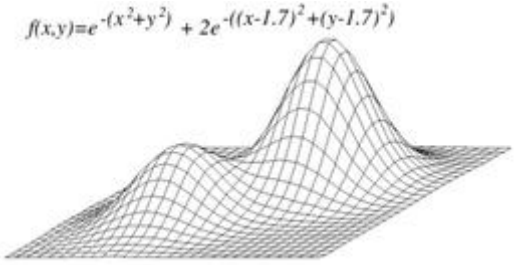

Figure 1. A problem with two hills

In discrete cases each candidate of the search space is identified with a vertex. A Hill climbing search will move from vertex to vertex. An evaluation of $f(\underline{x})$ is made at each vertex and is preferred if the new solution is better than the previous one. It ends when a locally optimal solution is obtained.

Some particular criteria generate different types of algorithms:

1. **simple hill climbing**: the first closer node is chosen
2. **Steepest ascent hill climbing** all neighbors are compared and the closest to the solution is chosen.
3. **Best-first search**, tried all possible extensions of the current path instead of only one.
4. **Stochastic hill climbing** a neighbor is selected at random. If the gain is sufficiently large the move is accepted or another one is selected and analyzed.
5. Coordinate descent: A search I developed along a coordinate direction at the current point at each iteration.
6. **Random-restart hill climbing** or **Shotgun hill climbing**. A hill-climbing is developed using a random initial start. The optimal solution is recorded and the procedure is repeated for a number of times. The best of the finally accepted solutions is the optimal solution.

Some of these algorithms are considered as stochastic. We refer to Stochastic optimization (SO) for identifying optimization methods that generate and use random variables in the solution of a problem. SO methods generalize deterministic methods existing for deterministic problems. We can identify two large classes of problems:

1. The random variables are included in the formulation of the optimization problem itself involving random objective functions or random constraints.
2. Methods that use in some stage random iterates.

Some SO methods use random iterations for solving stochastic problems, combining both meanings type of SO methods. Hence we can develop LS using SO, Stochastic LS. Stochastic local search is a successful technique in diverse areas of combinatorial optimization and is predominantly applied to hard problems. When dealing with individual instances of hard problems, gathering information about specific properties of instances in a pre-processing phase is helpful for an appropriate parameter adjustment of local search-based procedures.

The introduction of randomness into the search-process usually accelerates the process of finding the optima. It improves the method making it less sensitive to modeling errors and search procedures. For example injecting randomness may enable escaping from being trapped in a local minimum and not identifying a global optimum.
We can use SO in developing LS. They perform well when compared with the standard LS. Usually in problem-specific evaluations they only give some guidance information regarding what is the best direction for choosing the next LS-step, looking for escaping from getting stuck in local optima. The introduction of some randomness in the search allows escaping form the so called "dead ends." These procedures conforms the class of Stochastic LS (SLS). The Decision Maker makes al description of SLS fixing:

- An initial probability distribution function F_D on S.
- Taking *PD(x)=P(that candidate s∈S under* F_D*.)*

For each $s \in S$, a local search probability distribution $L(x)$ on $\mathcal{N}(s)$ is determined .Then $P_{L(s)})(s^*)=P(s^*)$ is selected in the LS- step starting with *s*)

Mathematically we are fixing that

$$P_{L(s)} = \begin{cases} lies\ in\ (0,1) if\ s^* \in \mathcal{N}(s) \\ 0\ if\ s^* \notin \mathcal{N}(s) or\ \mathcal{N}(s) = \varnothing \\ 1\ if\ there\ is\ a\ failure \end{cases}$$

Hence the generic local search algorithm for a problem is implemented by the pseudo code
.

A pseudo code for SLS
INPUT the problem
Choose an initial candidate solution $s \in S$ according to distribution F_D
WHILE (*s* is not the optimal solution or *s* is not a failure) DO
Replace s by one of its neighbors $s^* \in \mathcal{N}(s)$ selected according to the distribution $L(x)$
END-WHILE

Generally it is impossible to establish if *s* is an optimal solution and a function *g* is used for measuring its goodness. Then *g* becomes optimal if *s* is an optimal solution. Hence if the value g*(s)* remains unchanged for some evaluations this might support stopping. Also, the definition of *L(s)* might be based on the *g*-values generated by *s* and its neighbors. In the simplest implementations it might be the case that s is a failure; that is $P_{L(s)}(s) = 1$.

Usually SLS is implemented by selecting one general algorithm. The involved parameters are calibrated for enhancing a good behavior in terms of accuracy or time consumption.

3.2. Metaheuristics

3.2.1. Reasons for using a Metaheuristic

In practice, it is sometimes almost impossible to obtain global solutions in the strict sense. In many problems, it is sufficient having a local optimum or at least being close to i: Say that we are interested in solution s such that the objective function is "as high (small) as possible".

Metaheuristics are popularly used for solving COP`s a discrete search-space because usually the search-space of candidate solutions grows more than exponentially with the problem the size. Hence an exhaustive search for the optimal solution is generally infeasible. This is particularly true in multidimensional combinatorial problems.

In discrete search spaces are of current used for solving COP`s simulated annealing, genetic algorithms, ant colony optimization, scatter search and tabu search. for real-valued search-spaces the tools of optimization is deriving the gradient of the objective function and employing gradient descent or a quasi-Newton method. The most used Metaheuristics are particle swarm optimization, differential evolution and evolution strategies.

Some mathematically based studies of the performance of the Metaheuristics are scarce. Convergence results for simulated annealing have been derived modeling it as a sequence of homogeneous Markov chains or as a single inhomogeneous Markov chain see Johnson and Jacobson, 2002..For genetic algorithms we have the so called. Holland's schema theorem, Holland (1975); the result of the thesis of Rechenberg (1971) on evolution strategies; the recent paper of Trelea (2003) sustaining how to develop the analysis of particle swarm optimization.

The study of the performance and convergence aspects of Metaheuristic optimizers are often sustained by empirical research. Wolpert and Macready (1997) derived that all optimizers perform similarly on the average. This results identified as " no free lunch theorems" are unimportant because the hypothesis sustaining them usually do not hold in many practical situations. The main problem is that here is not a Metaheuristics guaranteeing tat the optimum or even a satisfactory near-optimal solution is obtained. They can face problems on which the performance is poor. So it is necessary gaining experience on each class of problems.

We will consider Simulated Annealing and Tabu Search in our reseearch

3.2.2. Simulated annealing

Simulated annealing (SA) is often used to solve not only discrete optimization problems but also in continuous optimization solving it may provide adequate solutions to problems belonging to the NP-complete class of problems. Simulated annealing mimics the process undergone by misplaced atoms in a metal when it is heated and then slowly cooled. The process of physical annealing with solids, in which a crystalline solid is heated and then allowed to cool very slowly until it achieves its minimum lattice energy state, say be free of crystal defects. Then the final configuration results in a solid with such superior structural integrity. Simulated annealing establishes the connection between thermodynamic behavior and the search for global minima for a discrete optimization problem and provides an algorithmic means for exploiting this connection. The key algorithmic issue SA is that it provides a means for avoiding being trapped by y local optima by accepting *hill-climbing* moves worsening the objective function value

SA is easy to implement and its, convergence properties and its use of hill-climbing moves to escape local optima have made it very popular. This technique is unlikely to find the global *optimal* solution but often it finds good solutions even in the presence of noisy data. At each iteration of a SA algorithm, applied to a discrete optimization problem, the objective function generates values for two solutions (the current solution and a new randomly selected solution). They are compared. Improving solutions are always accepted but a fraction of non-improving solutions are accepted looking for escaping from the curse of the local optima in search of global optima. The probability of accepting non-improving solutions depends on a temperature parameter T. It is non-increasing with each iteration of the algorithm. As T is decreased to zero, hill climbing moves occur less frequently. The solution distribution is described by a no-homogeneous Markov chain modeling the behavior of the algorithm. It is proved that the algorithm converges to a form in which all the probability is concentrated on the set of globally optimal solutions, provided that the algorithm is convergent. If not it converges to a local optimum.

SA is based on the Metropolis acceptance criterion (Metropolis et al., 1953). It models how a thermodynamic system moves from the current state (solution) s to a candidate solution s* in which the energy content is

being minimized. Denote by $t_k > 0$ the temperature parameter at (outer loop) iteration k. It must satisfy that $\lim_{k\to\infty} t_k = 0$. The candidate solution, is accepted as the current solution based on the acceptance probability

$$P_k(accept\ s^*) = \begin{cases} e^{-\frac{f(s^*)-f(s)}{t_k}\ if\ f(s^*)-f(s)=\Delta_{s^*.s}\ >0} \\ 1\ if\ \Delta_{s^*.s}\ \leq 0 \end{cases}$$

The equilibrium of the system follows the Boltzmann distribution, which is described by the probability that the system be in state $s \in S$ with energy $f(s)$ at temperature T such that

$$P_k(the\ system\ is\ in\ state\ s\ at\ temperature\ T) = \frac{e^{-\frac{f(s)}{t_k}}}{\sum_{w\in S} e^{-\frac{f(w)}{t_k}}}$$

and if

$$\sum_{s^*\in\mathcal{N}(s)} P_k(s,s^*) = 1, P_k(s,s^*) = P(genereting\ s^* \in \mathcal{N}(s))$$

We have a non-negative square stochastic matrix with transition probabilities

$$\forall s \in S, k = 0,1,2,..,: \pi_k(s,s^*) = \begin{cases} P_k(s,s^*)e^{-\frac{\Delta_{s^*.s}}{t_k}\ if\ s^*\in\mathcal{N}(s)and\ s^*\neq s} \\ 0\ if\ s^*\notin\mathcal{N}(s) \\ 1 - \sum_{\{s^{**}\in\mathcal{N}(s)\ and\ s^{**}\neq s\}} \pi_k(s,s^{**})\ if\ s^* = s \end{cases}$$

These are transition probabilities defining a sequence of solutions generated from an inhomogeneous Markov chain.

A pseudo code for implementing SA

Fix an initial solution $s \in S$ and an initial temperature $t_0 \geq 0$, $k=0$
Fix a temperature cooling parameter A
Fix a repetition schedule M_k that defines the number of iterations executed for temperature a t_k
Repeat
Set repetition counter $m = 0$
Repeat
Generate a solution $s^* \in \mathcal{N}(s)$
Calculate $\Delta_{s^*.s}$
$If\ \Delta_{s^*.s}\ \leq 0\ then\ s = s^*$

$If\ \Delta_{s^*.s}\ > 0\ perform\ a\ random\ experiment\ with\ probability\ e^{-\frac{\Delta_{s^*.s}}{t_k}}$
$m = m + 1$
Until $m=M_k$
$k = k + 1$
Until stopping criterion is met□

The total number of iteration of this algorithm is $M_0+M_1+....+M_k$.

3.2.3. Tabu Search

The Tabu Search (TS) method was proposed in 1986 by Glover to solve COP`s. Two principles of the TS method are the neighborhood search approach and the tabu list (TL), respectively. The simplest is called naïve tabu search (NTS) that is usually trapped by local solutions. The convergence analysis of the conventional TS is derived by Glover-Hanafi (2001) and Hanafi (2000).

The general TS consists in constructing from a solution s another one s^* checking whether one should stop there or perform another search. It uses the corresponding neighborhood $\mathcal{N}(s)$ of the feasible solution s for

looking for an improvement .For improving the efficiency of the search the procedure keeps track on local information and on the exploration process.

The method considers that S have k strictly local minima and is divided into k regions *S(i)* . Each region has w members and contains only one local minimum, which must not be located at the boundary. A finite sub-space of $S^* \subset S$ is randomly determined, $/S^*/ = m < n$. Take a finite sequence $s(i) = \{s_0, i\}$, $i = 1,2,\ldots,p$, as a collection of solution movements, s_0, consisting of p solutions to reach the global minimum ($k < p$). Denote by *t(s)* the time consumed to visit a solution s in the search space. Usually $t(s)=t(s^*)>0$, when moving in *S*. for $i = 1,2,\ldots,n$. The term Iteration counts the number of iterations, how many solutions in Swere already visited. Count do a similar job with the cumulative search round of sub-space explorations, how many sub-spaces in Swere already explored entirely. Another involved variable is the Back-tracking (BT) mechanism which controls the use of previously visited local optima the Tabu-List for generating a new starting point different from the one in hand. Finally the TS needs of an adaptive search radius mechanism (AR) which allows reducing the access time to local optima.

As there are non-improving moves is recommended to avoid the risk of visiting again a solution and of generating cycling. Then the BT allows forbidding moves which might lead to recently visited solutions. Now the structure of $\mathcal{N}(s)$ will depend upon the itinerary and hence upon the iteration *k*. Therefore we may refer to $\mathcal{N}(s,k)$ and s^* as the best solution found so far and *k* the *Iteration counter*.

Some times we look for efficiency using several lists T_r simultaneously. Then some constituents t_r will be given a tabu status to indicate that these constituents are currently not allowed to be involved in a move. Generally the tabu status of a move is a function of the tabu status of its constituents which may change at each iteration. That is, $t_r(i,m) \in T_r$ $(r=1,\ldots,t)$. A move *m* (applied to a solution *s*) will be a tabu move if all conditions are satisfied. Sometimes we accept m in spite of its status by fixing aspiration level a(i,m) which is better than a certain fixed threshold value *A(i,m)* which identifies some set of preferred values for a function *a(i,m)*. This is formalized writing $a_r(i,m) \in A_r(i,m)$ $(r=1,\ldots,a)$. If at least one of these conditions is satisfied by the tabu move *m* applied to *s*, then m will be accepted.

In some model is convenient modifying the objective function f another function f^* for intensifying and diversifying the search. Formally the DM is able to define it as $f^*=f+Intensification+Diversification$. In order to intensify the search in promising regions, we may decide returning to one of the best previously obtained solutions. Diversification is instrumented by using different random restarts or by penalizing frequently performed moves or solutions often visited.

A pseudo code for implementing TS

Step 1. Fix an initial solution $s \in S$. Set $s^*=s$ and $k=0$.
Step 2. Set $k=k+1$ and generate a subset V^* of solutions in $\mathcal{N}(s,k)$ such that:
  either one of the tabu conditions $tr(i,m) \in Tr$ is violated $(r=1,\ldots,t)$
  or at least one of the aspiration conditions $a_r(i,m) \in A_r(i,m)$ holds $(r=1,\ldots,a)$.
Step 3. Choose a best $s^*=s$, $m \in V^*$ (with respect to *f* or f^*) and set $s=s^*$.
Step 4. If *f(s)* is preferred to $f(s^*)$ then set $s^*=s$.
Step 5. Update tabu and aspiration conditions.
Step 6. If a stopping condition is met then stop. Else go to Step 2.□

In TS some usually defined stopping conditions are:
- $\mathcal{N}(s,k+1) = \varnothing$
- *k* is larger than the maximum number of iterations allowed
-The number of iterations since the last improvement of s^* is larger than a specified number *K*
- We have evidence that a optimum solution has been obtained.

The efficiency of the TS depends mostly on the modeling which relies on the parameters.

.

4. THE STUDY OF CONSENSUS RANKING FOR JOURNALS

4.1 The problem of ranking journals

The need for a complete journal ranking is growing as researchers and academicians obtains financing, promotion, or grants as a function of good evaluation of their research outputs. The number of academic journals is growing as a result; it is certainly more and more difficult assessing the quality of research through the journal where a result has been published. Hence is needed to have an unabridged evaluation of papers for any journal. Develop a complete ranking of them is useful for all these reasons.

Thomson Reuters ISI Web of Science (2011 is the most prestigious source of quantifiable bibliometric Research Assessment Measures (RAM). It constitutes a benchmark for comparing other general databases, such as SciVerse Scopus, Google Scholar and Microsoft Academic Search, social science open access repositories. When impact factors and other citations data are used without appropriate care misleading, unintended inferences may be drawn. They are not more than a descriptive statistics to capture journal impact and performance, and as they are not based on a mathematical model, no sound inferences can de done.

Bibliometrics has been increasingly using new metrics such as the h-index, EigenFactor, SJR, and SNIP.

In practice only a relatively small percentage of papers from many journals are included in the top databases and it is not clear how the system bases the selections. A main aspect of the systems is that the ratings are not based on bibliometric data at the journal level, but on a somewhat historical expert evaluation at level of the papers. A key role for evaluating a paper is played by citations in top rated journals. nevertheless it is not a reliable measure of intellectual achievements of the author because are equally considered positive and negative and neutral citations. The existent ranking on relies peer assessment of journal quality provided by a few available citation indexes, must of them elitist.. Seglen (1997) cautioned against using impact factors of journals to evaluate scientific research. It seem s more reliable o evaluate he author, see Hirsch (2005.

We developed a research for developing a consensus ranking that allows evaluating more or less selective aspects of journals. We used the 299 leading journals in Economics analyzed by Chen et al (2012). Our man objective was to consider how the SLS works inconsnesus ranking. The involved RAM`s are

(1) the 2-year impact factor including journal self citations (2YIF): Total citations in a year to papers published in a journal in the previous 2 years / Total papers published in a journal in the previous 2 years".
(2) the 2-year impact factor excluding journal self citations (2YIF*): 2YIF excluding self citations.
(3) the 5-year impact factor including journal self citations (5YIF): "Total citations in a year to papers published in a journal in the previous 5 years / Total papers published in a journal in the previous 5 years."
(4) Immediacy, or zero-year impact factor including journal self citations (0YIF): "Total citations to papers published in a journal in the same year / Total papers published in a journal in the same year."
(5) 5YIF Divided by 2YIF (5YD2): "5YD2 = 5YIF / 2YIF".
(6) Eigenfactor (or Journal Influence): Use the Eigenfactor algorithm (www.eigenfactor.org/methods.htm) ranks journals according to citations and the length of time that researchers are logged on to a journal's website.
(7) Article Influence: "Eigenfactor score divided by the fraction of all articles published by a journal."
(8) IFI: is the Impact Factor Inflation (IFI) given by "IFI = 2YIF / 2YIF*".
(9) H-STAR: "H-STAR = [(100-HS) - HS] =(100-2HS)". If HS = 0 (minimum), 50 or 100 (maximum) percent.
(10) 2Y-STAR: If 2YS = journal self citations over the preceding 2-year period,
"2Y-STAR = [(100-2YS) – 2YS] = (100-2(2YS))". If 2YS = 0 (minimum), 50 or
100 (maximum) percent.
(11) Escalating Self Citations (ESC): "ESC = 2YS – HS = (H-STAR – 2YSTAR)
/ 2".
(12) C3PO: "C3PO (Citation Performance Per Paper Online) = Total citations to a journal / Total papers published in a journal."
(13) h-index: "h-index = number of published papers, where each has at least h citations."
(14) PI-BETA:PI-BETA(= Papers Ignored (PI) - By Even The Authors (BETA)),.
(15) CAI: "CAI = (1 - PI11BETA)(Article Influence)".

The computing experiments were performed on a Intel® Xeon™ 2,40 GHz CPU with 2GB of RAM and the programs used MATLAB tool kits.

4.2 The experimentation with SA

The parameters used in the Monte Carlo experiments with SA were:

i) the initial temperature t_0,
ii) the threshold $M_k = M$, for any k, for the number of repetitions of solution cycling before diminishing the temperature,
iii) the cooling parameter A.

We used two replica for each combination. Each parameter setting is used for computing solutions in *1,000* independent trials. The values of the paramtrere were
i) the initial temperature $t_0 = 100, 10$,
ii) the threshold $M = 100, 1000$,
iii) the cooling parameter $A = 0{,}90 , 0{,}10$.

The comparison of the means of the optimal value computed is developed using ANOVA for SA.

Table 1. ANOVA of the experiments with Simulated Annealing

Source of variation	*Degrees of freedom*	*Mean square*	*F*	*p-value*
Temperature (T)	*1*	*0,3393*	*1,8355*	*0,1756*
Maximum number of repetitions (M)	*1*	*0,4729*	*2,5537*	*0,07792*
Cooling parameter (A)	*1*	*1,4780*	*7,9805*	*0,0003*
T×M	*1*	*3,3037*	*17,8385*	*3,072e-05*
T×A	*1*	*1,9467*	*10,511*	*4,596e-14*
M×A	*1*	*9,6783*	*52,2571*	*5,6351e-13*
T×M×A	*1*	*3,4796*	*18,7882*	*9,3266e-15*
Residual	*4992*	*0,1852*		

The experiments sustain, see the ANOVA for SA, that only changes in the cooling parameters generate significantly different means. All the interactions generate significant differences .

4.3. The experimentation with TS

The parameters used in the Monte Carlo experiments with TS were:

iv) the initial search radius (*R*),
v) the number of neighborhood members *(#nm)*,
vi) the number of repetitions of solution cycling before back-tracking (*n_re_max*),
vii) the *k*th backward solution selected by the back-tracking mechanism (*kth-bs*)
viii) the percentage of search radius reduction compared to the radius before adaptation (*DF*).

We used two replica for each combination. Each parameter setting is carried out in *1,000* independent trials. It starts assigning the mean ranks of each item. The searches stopped when hold one of the conditions
TS1) the maximum search round of 10,000
TS2) the cost function sufficiently small.

We implemented if by fixing that *R(old)* and *R(new*) are the search radius before and after adaptation. The values of *DF*,the factor of radius reduction, were *10%* and *50%* of the current radius and *R = 2,0%, 10,0%.; #nm=10, 50 ; n_re_max =5 and 20, kth-bs = –2, .-5};*
The adaptive radius scheme is set to have three steps of reduction as:
i) if [cost function < *0,1*] then *[R(new) = R(old)/DF];*
ii) if [cost function < *0,001*] then [*R(new) = R(old)/DF*];
iii) if [cost function < *0,001*]

Table 2. ANOVA of the experiments with Tabu Search

Source of variation	*Degrees of freedom*	*Mean square*	*F*	*p-value*
R	*1*	*3,2789e-03*	*0,1296*	*>5e-02*
DF	*1*	*0,1093*	*4,3212*	*>1e-02*
#nm	*1*	*8,797e-02*	*3,4769*	*>1e-02*
n_re_max	*1*	*0,1153*	*4,5591*	*>1e-02*
kth-bs	*1*	*2,2612e-02*	*1,0328*	*<5e-02*
R×DF	*1*	*9,1606e-02*	*3,6208*	*<1e-02*
R×#nm	*1*	*4,317*	*170,6516*	*<2,2e-16*
R× n_re_max	*1*	*4,1448*	*163,8316*	*<2,2e-16*
R× kth-bs	*1*	*4,698*	*185,7116*	*<2,391e-12*
DF×#nm	*1*	*1,2779*	*50,5112*	*2,391e-12*
DF× n_re_max	*1*	*1,3527*	*53,47*	*<2,391e-12*
DF× kth-bs	*1*	*0,4495*	*17,770*	*<5e-02*
#nm× n_re_max	*1*	*1,30874*	*51,7312*	*2,391e-12*
#nm× kth-bs	*1*	*7,572e-02*	*2,9931*	*>1e02*
n_re_max× kth-bs	*1*	*7,2593e-02*	*2,8693*	*>1e02*
R×DF×#nm	*1*	*1,2695*	*50,181*	*<2,391e-12*
R×DF× n_re_max	*1*	*1,311*	*51,84*	*2,391e-12*
R×DF× kth-bs	*1*	*4,067*	*160,79*	*<2,2e-16*
R×#nm× n_re_max	*1*	*1,2648*	*49,9934*	*<2,391e-12*
R×#nm× kth-bs	*1*	*1,2651*	*50,0004*	*<2,391e-12*
R×#nm×n_re_max kth-bs	*1*	*1,2390*	*48,9741*	*<2,391e-12*
DF×#nm× n_re_max	*1*	*4,5814*	*181,2084*	*<2,2e-16*
DF× n_re_max× kth-bs	*1*	*4,7030*	*185,8886*	*2,2e-16*
DF×#nm× kth-bs	*1*	*4,1202*	*162,8530*	*<2,2e-16*
#nm× n_re_max× kth-bs	*1*	*4,2476*	*167,8931*	*<2,2e-16*
R×DF×#nm× n_re_max× kth-bs	*1*	*4,5227*	*181,7516*	*<2,2e-16*
Residual	*4979*	*0,0253*		

The experiments sustain, see the ANOVA for TS that the values of the parameters are unimportant in terms of the mean of the solutions derived as the corresponding sources are non-significant. All the interactions generate significant differences excepting at the second level for *R×DF* , *DF× kth-bs*, *#nm× kth-bs* and *n_re_max× kth-bs*. Therefore once *R* is fixed the values of DF are non-influent in the solution obtained. Similarly DF has no interaction witht *kth-bs* and *kth-bs* with *#nm* and *n_re_max* ·

4.4. Study of the computing times

Let us consider a variable *Y* describing the run time and denote by $F_{RT}(t)$ its distribution function. It is continuous. Each Metaheuristic *M* has a distribution function indexed by the set of parameters fixed for performing SLS. Take α as these parameters. Hence we deal with $F_{RT}(t/M,\alpha)=P(Y\leq t)$. This probability measures our likelihood that the algorithm finds the solution with run time t in a single run. We estimate it using the empirical distribution function

$$F_{n,RT}(t|M,\alpha)=\frac{\sum_{v=1}^{numberofruns} I(Y_v\leq t)}{numberofruns}, \qquad I(q)=\begin{cases}1\ if\ q\ is\ true\\ 0\ if\ q\ is\ false\end{cases}$$

Clearly $Y_1,\ldots,Y_{numberofruns}$ are independent variables ,as the runs are independent. Our interest is to establish if the DF varies with α for a certain *M*.

We perform *1000* runs for each α hence we deal with a large sample of independent and identically distributed variables. This fact supports, see Hollander-Wolfe (1999) that $F_{n,RT}(t|M,\alpha)\rightarrow F_{RT}(t|M,\alpha)$ and that the Kolmogorov-Smirnov test can be used for testing whether if $H0$: $F_{RT}(t|M,\alpha)=F_{RT}(t|M,\alpha^*)$ or not. We perform the test considering the set $\Xi=\{\alpha\}$. A test of equality of the different instances of the algorithm is performed computing

$$D_{multiple}=Sup_{t,\alpha,\alpha^*}\left|F_{n,RT}(t|M,\alpha)-F_{n,RT}(t|M,\alpha^*)\right|,\alpha\neq\alpha^*;\ \alpha,\alpha^*\in\Xi$$

We accept for a certain significance level p that all the $F_{n,RT}(t|M,\alpha)`s$ are equal if $D_{multiple}>D_{KS}(1000,1000,p)$. See Birbaum-Hall test, see Conover (1980). In case of detecting that the FD`s of a pair of parameters set are different a pairwise study must be performed using

$$D_0=Sup_t\left|F_{n,RT}(t|M,\alpha)-F_{n,RT}(t|M,\alpha^*)\right|,\alpha\neq\alpha^*;\ \alpha,\alpha^*\in\Xi$$

As the sample size is large the asymptotic distribution of the Kolmogorov-Smirnov statistic is used.

The distribution of the times of TS were different for any pair of parameters sets wit *p*-values moving in (*0,003*, *0,007*).

The distributions for SA were unaffected by changing *T* for fixed values of *M* and *A*. The rest of the comparisons were significant with p-values moving in *(0,0006, 0,004).*

5. **CONCLUSIONS**

SA seems to be less sensitive to changes in the parameters. The cooling parameter A is the most influential one; hence the DM must be aware of that fact when implementing the procedure. The interaction among the parameters is important as the means associated differ . The distribution of the time for the temperature conditioned to a fixed set is the same.

TS is very sensitive for the interaction among the parameters. The distributions of any set of parameters differ significantly.

For determining a procedure for deriving a ranking consensus of the journals SA is recommended

Acknowledgments: the authors acknowledge the improvements of a previous version of his paper thanks to the recommending made by two anonymous referees.

RECEIVED AUGUST 2012
REVISED NOVEMBER 2012

REFERENCES

[1] AARTS, E., and J.K. LENSTRA (2003): **Local Search in Combinatorial Optimization**. Princeton University Press, Princeton.

[2] ARMSTRONG, R.D., COOK, W.D., and SEIFORD, L.M. (1982): Priority ranking and consensus formation: The case of ties. **Management Science** 28, 638-645.

[3] BEG, M.M.S., N. Ahmed (2003): Soft computing techniques for rank aggregation on

the world wide web. **World Wide Web: Internet and Web Information Systems,** 6, 5-22.

[4] BLACK, D. (1958): **The Theory of Committees and Elections**,Cambridge University Press, Cambridge, Michigan.

[5] BERGSTROM, C.T., J.D. West and M.A. Wiseman (2008), The Eigenfactor™ metrics, **Journal of Neuroscience**, 28, 11433–11434.

[6] CHANG, C.-L., M. MCALEER and L. OXLEY (2011): How are journal impact, prestige and article influence related? An application to neuroscience, **Journal of Applied Statistics**, 38, 2563-2573.

[7] CHANG, C-L., E. MAASOUMI and M. MCALEER (2012): Robust Ranking of Journal Quality: An Application to Economics. Submitted to Journal Quality: An Application to Economics. [**Working Paper or Technical Report last consulted July 12, 2012, available at http://www.ucm.es/info/icae/)**

[8] COHEN, R., L.KATZIR, and D. RAZ (2006): An Efficient Approximation for the Generalized Assignment Problem. **Information Processing Letters**, 100, 162-166.

[9] CONOVER J. (1980): **Practical nonparametric statistics**. Wiley, N. York,

[10] COOK, W.D. (2006): Distance-based and ad hoc consensus models in ordinal preference ranking. **European Journal of Operational Research** 172, 369–385

[11] COOK, W.D., B. GOLANY, M. KRESS, M. PENN and T. RAVIV (2005): Optimal Allocation of Proposals to Reviewers to Facilitate Effective Ranking. **Management Science** , 51, 655-661

[12] COOK, W.D., and KRESS, M. (1990): A data envelopment model for aggregating preference rankings. **Management Science** 36, 1302-1310.

[13] COOK, W. D., M. KRESS and L. M. SEIFORD (1996): A general framework for distance-based consensus in ordinal ranking models. **European Journal of Operational Research** 96, 392-397

[14] COOK, W.D., and SEIFORD, L.M. (1978): Priority ranking and consensus formation. **Management Science** 24, 1721-1732.

[15] COOK, W.D., and SEIFORD, L.M. (1982): The Borda-Kendall consensus method for priority ranking problems. **Management Sience** 28, 621-637.

[16] DWORK,C., R. KUMAR, M. NAOR, and D. SIVAKUMAR (2001): Rank aggregation methods for the web. **World Wide Web: Internet and Web Information Systems**, 1, 613–622,

[17] GLOVER, F. and HANAFI, S. (2001): Finite Convergence of Tabu Search. **Proc. MIC'2001-4th Metaheuristics Int. Conf, (**2001) 333-336

[18] GLOVER and LAGUNA (1997):. **Tabu Search**, Kluwer Academic Publishers.

[19] INGBER, L. (1993): Simulated Annealing: Practice Versus Theory. **Math. Comput. Modelling** 18, 29-57.

[20] **Proceedings of the National Academy of Sciences of the United States of America**, 102, 16569-15572 .

[21] HANAFI, S. (2000): On the Convergence of Tabu search. **J. Heuristics** 7, 47-58

[22] HOLLAND, J.H. (1975): **Adaptation in Natural and Artificial Systems**. University of Michigan Press, Michigan.

[23] HOLLANDER, M. and D.A.WOLFE (1999): **Nonparametric statistical methods**. Wiley Series in Probability and Statistics, Toronto

[24] HOOS, H.H. AND STUTZLE, T. (2005): **Stochastic Local Search: Foundations and Applications**, Morgan Kaufmann.

[25] ISI WEB OF SCIENCE (2011**): Journal Citation Reports, Essential Science Indicators**, Thomson Reuters ISI.

[26] JOHNSON, A.W. and JACOBSON, S.H. (2002): A class of convergent generalized hill climbing algorithms. **Applied Mathematics and Computation**, 125.

[27] KENDALL, M. (1962): **Rank Correlation Methods**. 3rd Edition, Hafner, New York.

[28] METROPOLIS, N., ROSENBLUTH, A., ROSENBLUTH, M., TELLER, A. AND TELLER, E. (1953): Equation of state calculations by fast computing machines. **Journal of ChemicalPhysics**, 21, 1087–1092.

[29] RECHENBERG, I. (1971**). Evolutionsstrategie – Optimierung technischer Systeme nach Prinzipien der biologischen Evolution** (PhD Thesis). ISBN 3-7728-0373-3.

[30] SEGLEN, P.O. (1997): Why the impact factor of journals should not be used for evaluating research, **BMJ: British Medical Journal**. 314, 498-502.

[31] STÜTZLE, T. and M. DORIGO (2001): Local Search and Metaheuristics for the Quadratic Assignment Problem. **Technical Report AIDA-01-01, Intellectics Group, Darmstadt University of Technology**.

[32] TRELEA, I.C. (2003): The Particle Swarm Optimization Algorithm: convergence analysis and parameter selection. **Information Processing Letters** 85, 317–325.

[33] WOLPERT, D.H. and W.G. MACREADY (1997): No free lunch theorems for optimization. **IEEE Transactions on Evolutionary Computation** 1, 67–82.