Scatter Search and Path Relinking: Advances and Applications

Fred Glover[a], Manuel Laguna[a], and Rafael Martí[b]

a Leeds School of Business, Campus Box 419, University of Colorado, Boulder, CO 80309-0419, USA. Fred.Glover@Colorado.edu and Manuel.Laguna@Colorado.edu

b Dpto. de Estadística e Investigación Operativa, Facultad de Matemáticas, Universitat de Valencia, Dr. Moliner, 50, 46100 Burjassot, Valencia, Spain. Rafael.Marti@uv.es

Latest Revision: June 26, 2002

Abstract — Scatter search (SS) is a population-based method that has recently been shown to yield promising outcomes for solving combinatorial and nonlinear optimization problems. Based on formulations originally proposed in the 1960s for combining decision rules and problem constraints, SS uses strategies for combining solution vectors that have proved effective in a variety of problem settings. Path relinking (PR) has been suggested as an approach to integrate intensification and diversification strategies in a search scheme. The approach may be viewed as an extreme (highly focused) instance of a strategy that seeks to incorporate attributes of high quality solutions, by creating inducements to favor these attributes in the moves selected.

The goal of this paper is to examine SS and PR strategies that provide useful alternatives to more established search methods. We describe the features of SS and PR that set them apart from other evolutionary approaches, and that offer opportunities for creating increasingly more versatile and effective methods in the future. Specific applications are summarized to provide a clearer understanding of settings where the methods are being used.

1. Introduction

Scatter search, from the standpoint of metaheuristic classification, may be viewed as an evolutionary (population-based) algorithm that constructs solutions by combining others. It derives its foundations from strategies originally proposed for combining decision rules and constraints in the context of integer programming. The goal of this methodology is to enable the implementation of solution procedures that can derive new solutions from combined elements in order to yield better solutions than those procedures that base their combinations only on a set of original elements. E.g., see the overview by Glover (1998).

The antecedent strategies for combining decision rules were first introduced in the area of scheduling, as a means to obtain improved local decisions. Numerically weighted combinations of existing rules, suitably restructured so that their evaluations embodied a common metric, generated new rules (Glover, 1963). The approach was motivated by the conjecture that information about the relative desirability of alternative choices is captured in different forms by different rules, and that this information can be exploited more effectively when integrated than when treated in isolation (i.e., by choosing selection rules one at a time). Empirical outcomes disclosed that the decision rules created from such combination strategies produced better outcomes than standard applications of local decision rules. The strategy of

creating combined rules also proved superior to a "probabilistic learning approach" that used stochastic selection of rules at different junctures, but without the integration effect provided by the combined rules (Crowston, et al., 1963).

The associated procedures for combining constraints likewise employed a mechanism of generating weighted combinations. In this case, nonnegative weights were introduced to create new constraint inequalities, called *surrogate constraints*, in the context of integer and nonlinear programming (Glover, 1965, 1968). The approach isolated subsets of (original) constraints that were gauged to be most critical, relative to trial solutions that were obtained based on the surrogate constraints. This critical subset was used to produce new weights that reflected the degree to which the component constraints were satisfied or violated. In addition, the resulting surrogate constraints served as source constraints for deriving new inequalities (cutting planes) which in turn provide material for creating further surrogate constraints.

Path Relinking has been suggested as an approach to integrate intensification and diversification strategies (Glover and Laguna, 1997) in the context of tabu search. This approach generates new solutions by exploring trajectories that connect high-quality solutions, by starting from one of these solutions and generating a path in the neighborhood space that leads toward the other solutions.

Recent applications of both methods (and of selected component strategies within them) that have proved highly successful are:

- **The Linear Ordering Problem** (Campos, Laguna and Martí)
- **The Bipartite Drawing Problem** (Laguna and Martí)
- **The Graph Coloring Problem** (Hamiez and Hao)
- **Capacitated Multicommodity Network Design** (Ghamlouche, Crainic and Gendreau)
- **The Maximum Clique Problem** (Cavique, Rego and Themido)
- **Assigning Proctor to Exams** (Ramalhinho, Laguna and Martí)
- **Periodic Vehicle Loading** (Delgado, Laguna and Pacheco)
- **Job Shop Scheduling** (Nowicki and Smutnicki)
- **The Arc Routing Problem** (Greistorfer)
- **Resource Constrained Project Scheduling** (Valls, Quintanilla and Ballestín)
- **Multiple Criteria Scatter Search** (Beausoleil)
- **Meta-Heuristic Use of Scatter Search via OptQuest** (Hill)
- **Pivot Based Search Integrated with Branch and Bound for Binary MIPs** (Løkketangen and Woodruff)
- **Scatter Search to Generate Diverse MIP Solutions** (Glover, Løkketangen and Woodruff)
- **Path Relinking to Improve Iterated Start Procedures** (Ribeiro and Resende)

A number of these applications are described in Section 4 where a collection of vignettes is presented. They provide a diverse range of settings where SS and PR have made useful contributions, and suggest the form of additional applications where similar successes may be anticipated.

2. Scatter Search

Scatter search is designed to operate on a set of points, called *reference points,* which constitute good solutions obtained from previous solution efforts. Notably, the basis for defining "good" includes special criteria such as diversity that purposefully go beyond the objective function value. The approach systematically generates combinations of the reference points to create new points, each of which is mapped into an associated feasible point. The combinations are generalized forms of linear combinations, accompanied by processes to adaptively enforce feasibility conditions, including those of discreteness (Glover, 1977).

The SS process is organized to (1) capture information not contained separately in the original points, (2) take advantage of auxiliary heuristic solution methods (to evaluate the combinations produced and to actively generate new points), and (3) make dedicated use of strategy instead of randomization to carry out component steps. SS basically consist of five methods:

1. A *Diversification Generation Method* to generate a collection of diverse trial solutions, using one or more arbitrary trial solutions (or seed solutions) as an input.

2. An *Improvement Method* to transform a trial solution into one or more enhanced trial solutions. (Neither the input nor the output solutions are required to be feasible, though the output solutions are typically feasible. If the input trial solution is not improved as a result of the application of this method, the "enhanced" solution is considered to be the same as the input solution.)

3. A *Reference Set Update Method* to build and maintain a *reference set* consisting of the b "best" solutions found (where the value of b is typically small, e.g., no more than 20), organized to provide efficient accessing by other parts of the solution procedure. Several alternative criteria may be used to add solutions to the reference set and delete solutions from the reference set.

4. A *Subset Generation Method* to operate on the reference set, to produce a subset of its solutions as a basis for creating combined solutions. The most common subset generation method is to generate all pairs of reference solutions (i.e., all subsets of size 2).

5. A *Solution Combination Method* to transform a given subset of solutions produced by the Subset Generation Method into one or more combined solutions. The combination method is analogous to the crossover operator in genetic algorithms although it should be capable of combining more than two solutions. (The combination processes proposed in the original SS paper included forms of "crossover" not envisioned in the GA literature until a decade later, and combination processes proposed since then, as in Glover (1994,1995) utilize principles and constructions that remain beyond the scope embraced by GA approaches.)

The basic procedure in Figure 1 starts with the creation of an initial reference set of solutions (*RefSet*). The Diversification Generation Method is used to build a large set of diverse solutions *P*. The size of *P* (*PSize*) is typically 10 times the size of *RefSet*. Initially, the reference set *RefSet* consists of b distinct and maximally diverse solutions from *P*. The solutions in *RefSet* are ordered according to quality, where the best solution is the first one in the list. The search is then initiated by assigning the value of TRUE to the Boolean variable *NewSolutions*. In step 3, *NewSubsets* is constructed and *NewSolutions* is switched to FALSE. For illustrative purposes we focus attention on subsets of size 2. Hence the cardinality of *NewSubsets* corresponding to the initial reference set is given by $(b^2-b)/2$, which accounts for all pairs of solutions in *RefSet*. (Special conditions are imposed on subsets of larger sizes to ensure a suitable composition is achieved while generating no more than a restricted number of these subsets.) The pairs in *NewSubsets* are selected one at a time in lexicographical order and the Solution Combination Method is applied to generate one or more solutions in step 5. If a newly created solution improves upon the worst solution currently in *RefSet*, the new solution replaces the worst and *RefSet* is reordered in step 6. The *NewSolutions* flag is switched to TRUE and the subset s that was just combined is deleted from *NewSubsets* in steps 7 and 8, respectively.

1. Start with $P = \emptyset$. Use the Diversification Generation Method to construct a solution x. If $x \notin P$ then add x to P (i.e., $P = P \cup x$), otherwise, discard x. Repeat this step until $|P| = PSize$. Build $RefSet = \{ x^1, \ldots, x^b \}$ with b diverse solutions in P.
2. Evaluate the solutions in *RefSet* and order them according to their objective function value such that x^1 is the best solution and x^b the worst. Make *NewSolutions* = TRUE.

while (*NewSolutions*) **do**

3. Generate *NewSubsets*, which consists of all pairs of solutions in *RefSet* that include at least one new solution. Make *NewSolutions* = FALSE.

while ($NewSubsets \neq \emptyset$) **do**

4. Select the next subset s in *NewSubSets*.
5. Apply the Solution Combination Method to s to obtain one or more new solutions x.

if (x is not in *RefSet* and $f(x) < f(x^b)$) **then**

6. Make $x^b = x$ and reorder *RefSet*.
7. Make *NewSolutions* = TRUE.

end if

8. Delete s from *NewSubsets*.

end while

end while

Figure 1. Outline of basic scatter search for a minimization objective.

This basic design can be expanded and improved in different ways. The SS methodology is very flexible, since each of its elements can be implemented in a variety of ways and degrees of sophistication. Different improvements and designs from this basic SS algorithm are given in Glover (1998), Glover, Laguna and Martí (1999 and 2000), Laguna (2000) and Laguna and Armentano (2001).

3. Path Relinking

One of the main goals in any search method is to create a balance between search intensification and search diversification. Path relinking has been suggested as an approach to integrate intensification and diversification strategies (Glover and Laguna, 1997). Features that have been added to Scatter Search, by extension of its basic philosophy, are also captured in the Path Relinking framework. This approach generates new solutions by exploring trajectories that connect high-quality solutions — by starting from one of these solutions, called an *initiating solution*, and generating a path in the neighborhood space that leads toward the other solutions, called *guiding solutions*. This is accomplished by selecting moves that introduce attributes contained in the guiding solutions.

The approach may be viewed as an extreme (highly focused) instance of a strategy that seeks to incorporate attributes of high quality solutions, by creating inducements to favor these attributes in the moves selected. However, instead of using an inducement that merely encourages the inclusion of such attributes, the path relinking approach subordinates other considerations to the goal of choosing moves that introduce the attributes of the guiding solutions, in order to create a "good attribute composition" in the current solution. The composition at each step is determined by choosing the best move, using customary choice criteria, from a restricted set – the set of those moves currently available that incorporate a maximum number (or a maximum weighted value) of the attributes of the guiding solutions.

The approach is called path relinking either by virtue of generating a new path between solutions previously linked by a series of moves executed during a search, or by generating a path between solutions previously linked to other solutions but not to each other. Figure 2 shows two hypothetical paths (i.e., a sequence of moves) that link solution A to solution B, to illustrate relinking of the first type. The solid line indicates an original path produced by the "normal" operation of a procedure that produced a series of moves leading from A to B, while the dashed line depicts the relinking path.

The paths are different because the move selection during the normal operation does not "know" where solution B lies until it is finally reached, but simply follows a trajectory whose intermediate steps are determined by some form of evaluation function. For example, a commonly used approach is to select a move that minimizes (or maximizes) the objective function value in the local sense. During path relinking, however, the main goal is to incorporate attributes of the guiding solution (or solutions) while at the same time recording the objective function values.

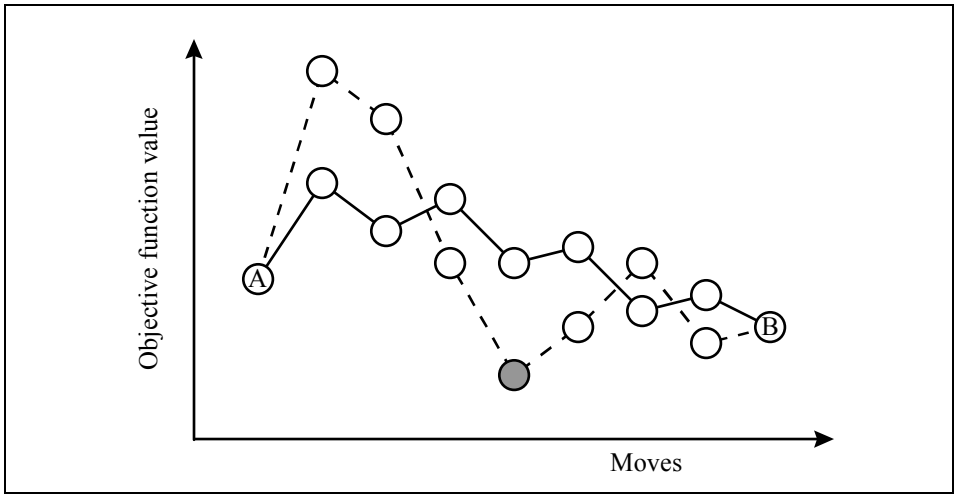


Figure 2. Path relinking illustration.

The effort to represent the process in a simple diagram such as the one preceding creates some misleading impressions, however. First, the original (solid line) path, whch is shown to be "greedy" relative to the objective function, is likely to be significantly more circuitous along dimensions we are not able to show, and by the same token to involve significantly more steps (intervening solutions) – an aspect not portrayed in Figure 2. Second, because the relinked path is not governed so strongly by local attraction, but instead is influenced by the criterion of incorporating attributes of the guiding solution, it opens the possibility of reaching improved solutions that would not be found by a "locally myopic" search. Figure 2 shows one such solution (the darkened node) reached by the dotted path. Beyond this, however, the relinked path may encounter solutions that may not be better than the initiating or guiding solution, but that provide fertile "points of access" for reaching other, somewhat better, solutions. For this reason it is valuable to examine neighboring solutions along a relinked path, and keep track of those of high quality which may provide a starting point for continued search.

The incorporation of attributes from elite parents in partially or fully constructed solutions was foreshadowed by another aspect of scatter search, embodied in an accompanying proposal to assign preferred values to subsets of *consistent* and *strongly determined* variables. The theme is to isolate assignments that frequently or influentially occur in high quality solutions, and then to introduce compatible subsets of these assignments into other solutions that are generated or amended by heuristic procedures. (Such a process implicitly relies on a form of frequency-based memory to identify and exploit variables that qualify as consistent.)

Multiparent path generation possibilities emerge in path relinking by considering the combined attributes provided by a set of guiding solutions, where these attributes are weighted to determine which moves are given higher priority. The generation of such paths in neighborhood space characteristically "relinks" previous points in ways not achieved in the previous search history, hence giving the approach its name. This multiparent Path Relinking approach generates new elements by a process that emulates the strategies of the original Scatter Search approach at a higher level of

generalization. The reference to neighborhood spaces makes it possible to preserve desirable solution properties (such as complex feasibility conditions in scheduling and routing), without requiring artificial mechanisms to recover these properties in situations where they may otherwise become lost.

The PR approach benefits from a **tunneling strategy** that often encourages a different neighborhood structure to be used than in the standard search phase. For example, moves for Path Relinking may be periodically allowed that normally would be excluded due to creating infeasibility. Such a practice is protected against the possibility of becoming "lost" in an infeasible region, since feasibility evidently must be recovered by the time the guiding solution is reached.

A natural variation of path relinking occurs by using **constructive neighborhoods** for creating offspring from a collection of parent solutions. In this case the guiding solutions consist of subsets of elite solutions, as before, but the initiating solution begins as a partial (incomplete) solution or even as a null solution, where some of the components of the solutions, such as subsets of free variables, are not yet assigned. The use of a constructive neighborhood permits such an initiating solution to "move toward" the guiding solutions, by a neighborhood path that progressively introduces elements contained in the guiding solutions, or that are evaluated as attractive based on the composition of the guiding solutions.

4. SS / PR Vignettes

This section provides a collection of "vignettes" that briefly summarize applications of SS and PR in a variety of settings. These vignettes are edited versions of reports by researchers and practitioners who are responsible for the applications. A debt of gratitude is owed to the individuals whose contributions have made this summary possible.

4.1 The Linear Ordering Problem

Given a matrix of weights $E = \{ e_{ij} \}_{m \times m}$, the Linear Ordering Problem (LOP) consists of finding a permutation p of the columns (and rows) in order to maximize the sum of the weights in the upper triangle. In mathematical terms, we seek to maximize:

$$C_E(p) = \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} e_{p(i)p(j)} .$$

where $p(i)$ is the index of the column (and row) in position i in the permutation. In the LOP, the permutation p provides the ordering of both the columns and the rows. The equivalent problem in graphs consists of finding, in a complete weighted graph, an acyclic tournament with a maximal sum of arc weights. In economics, the LOP is equivalent to the so-called *triangulation problem for input-output tables* (Reinelt, 1985).

Campos et al. (1999) propose a solution method for the linear ordering problem based on the scatter search template in Glover (1998). The procedure combines the following elements:

a) Diversification Generator
b) Improvement Method
c) Reference Set Update Method
d) Subset Generation Method
e) Solution Combination Method

where a), b) and e) are context dependent and c) and d) are "generic" elements.

The authors developed and tested 10 **Diversification Generation Methods**. These methods included a completely random method, several versions of GRASP, a deterministic method that disregards the objective function, and a method using frequency-based memory as proposed in tabu search. The diversification approach using TS frequency-based memory was found to clearly outperform the competing methods.

The **Improvement Method** is based on the neighborhood search developed for the Tabu Search algorithm for the LOP in Laguna, Martí and Campos (1999).

The **Solution Combination Method** uses a min-max construction based on votes. The method scans each reference permutation to be combined, and uses the rule that each reference permutation votes for its first element that is still not included in the combined permutation (referred to as the "incipient element").

In a set of **Computational Testing Experiments**, the authors compare the performance of two variants of the scatter search implementation with three methods: Chanas and Kobylanski (CK, 1996), Tabu Search (TS, Laguna, Martí and Campos, 1998) and a greedy procedure especially designed for the LOP. The scatter search procedures were tested on four sets of instances.

The tabu search method and the two scatter search instances dominate the other approaches in terms of solution quality. The TS method is the fastest of the high quality methods, running 3 to 5 times faster than the first scatter search variant, but the scatter search variants give the best overall solution quality, indicating their value where quality is the dominant consideration.

4.2 The Bipartite Drawing Problem

The problem of minimizing straight-line crossings in layered bipartite graphs consists of aligning the two shores V_1 and V_2 of a bipartite graph $G = (V_1, V_2, E)$ on two parallel straight lines (layers) such that the number of crossing between the edges in E is minimized when the edges are drawn as straight lines connecting the end-nodes. The problem is also known as the *bipartite drawing problem* (BDP).

The main application of this problem is found in automated drawing systems, where drawing speed is a critical factor. Simple heuristics are very fast but result in inferior solutions, while high-quality solutions have been found with meta-heuristics that demand an impractical amount of computer time. Laguna and Martí (1999) propose a method that combines GRASP and Path Relinking to develop a procedure that can compete in speed with the simple heuristics and in quality with the complex meta-heuristics.

The hybrid procedure proposed for the BDP utilizes GRASP as the multistart method to be augmented, and stores a small set of high quality (elite) solutions to be used for guiding purposes.

In a set of **Computational Testing Experiments**, the authors compare the performance of the GRASP and Path Relinking implementations with two methods: the iterated barycenter (BC, Eades and Kelly, 1986) and a version of the Tabu Search algorithm (TS, Martí, 1996). The former is the best of the simple heuristics for the BDP (Martí and Laguna, 1997), while the later has been proven to consistently provide the best solutions in terms of quality. For these experiments 3,200 instances have been generated with the `random_bigraph` code of the Stanford GraphBase by Knuth (1993).

The first experiment shows that the best solution quality is obtained by the tabu search method, which is able to match the 900 known optimal solutions, while GRASP matches 750 and PR matches 866. However, in contrast to some other applications (such as the Linear Ordering problem previously described), TS employs more

computational time (15 seconds) than the other methods reported. GRASP performs quite well, considering its average percent deviation from optima of 0.44% achieved on an average of 0.06 seconds. Notably, path relinking achieves a significantly improved average percent deviation of 0.09% with a modest increase in computer time (0.28 seconds). Finally, iterated BC from 10 random starts turns in a substantially less attractive performance, with an average percent deviation of 3.43% achieved in 0.08 seconds.

The second experiment is devoted to sparse graphs. It is shown that the path relinking algorithm achieves the best average deviation of less than 1%. The computational effort associated with the PR variant is very reasonable (with a worst case average of 1.61 seconds).

A third experiment was performed to assess the efficiency of the proposed procedures in denser graphs (relative to the second experiment). The results show that the tabu search procedure outperforms both the BC and the GRASP variants. The average deviation from the best known values is 1.41% for the TS procedure, while PR obtains an average deviation of 8.96%, using similar computational time (i.e., 26 seconds for TS versus 22 seconds for PR).

The enhancements produced by path relinking suggests the potential merit of joining the PR guidance strategies with other multistart methods.

4.3 The Graph Coloring Problem

Graph k-coloring can be stated as follows: given an undirected graph G with a set V of vertices and a set E of edges connecting vertices, k-coloring G means finding a partition of V into k classes V_1, ..., V_k, called *color classes*, such that no couple of vertices $(u, v) \in E$ belongs to the same color class. Formally, $\{V_1, ..., V_k\}$ is a valid k-coloring of the graph $G=(V, E)$ if $\forall\ i \in [1..k]$ and $\forall\ (u, v) \in V_i$, $(u, v) \notin E$. The *graph coloring problem* (GCP) is the optimization problem associated with k-coloring. It aims at searching for the minimal k such that a proper k-coloring exists. This minimum is the chromatic number $\chi(G)$ of graph G.

Graph coloring has many real applications, e.g., timetable construction, frequency assignment, register allocation or printed circuit testing. There are many resolution methods for this problem: greedy constructive approaches (DSATUR, RLF), hybrid strategies (HCA for instance), local search metaheuristics (simulated annealing, tabu), neural network attempts, ... Despite the fact that the literature on graph coloring is always growing, there exists, to our knowledge, no approach relying on scatter search for the graph coloring problem. We summarize here such an experimental investigation following the scatter search template of Glover (1998).

Our **Diversification Generation Method** uses *independent sets* to build initial configurations. Color classes are built one by one by selecting vertices in a random order to insure diversity.

The **Improvement Method** is based on the tabu search algorithm of Dorne and Hao (1998). This algorithm iteratively changes the current color of a conflicting vertex to another one, until achieving a proper coloring. A tabu move leading to a configuration better than the best configuration found so far, within the same execution of the improvement method or within the overall scatter search procedure, is always accepted (*aspiration criterion*).

Although the **Reference Set Update Method** is usually a "generic" element of scatter search, we provide here the way configurations are compared in terms of *diversity*. This point is crucial since, in the context of graph coloring, the Hamming distance is not well suited to compare two configurations c_1 and c_2. The *distance* between c_1 and

c_2 is the minimum number of moves necessary to transform c_1 into c_2. The *fitness* of any configuration is naturally its number of conflicting edges.

The **Solution Combination Method** uses a generalization of the powerful *greedy partition crossover* (GPX), proposed by Galinier and Hao (1999) within an evolutionary algorithm. GPX has been especially developed for the graph coloring problem with results reaching, and sometimes improving, those of the best known algorithms for the GCP. Given a subset p generated by the subset generation method, the generalized combination operator builds the k color classes of the new configuration one by one. First, choose a configuration $c \in p$. Remove from c a minimal set of conflicting vertices such as c becomes a partial proper k-coloring. Next, fill in a free color class of the new configuration with all conflict-free vertices of the color class having maximum cardinality in c. Repeat these steps until the k color classes of the new configuration contain at least one vertex. Finally, to complete the new configuration if necessary, assign to each free vertex a color such that it minimizes the conflicts over the graph.

Computational Testing Experiments has been carried out on some of the well-known DIMACS benchmark graphs (Johnson and Trick, 1996). The scatter search procedure (SSGC) was compared with the generic tabu search (GTS) algorithm of Dorne and Hao (1998) together with the best-known methods available for the graph coloring problem: two local search algorithms based on particular neighborhoods and a distributed population-based algorithm (Morgenstern, 1996), and an hybrid method including a descent algorithm and a tabu procedure with various heuristics mixed with a greedy construction stage and the search for a maximum clique (Funabiki and Higashino, 2000).

The scatter search approach SSGC managed to reach the results of the best-known algorithms in quality (minimal number of colors used), except on the r1000.5 graph for which a 237-coloring has been published recently (Funabiki and Higashino, 2000). (The sophisticated algorithm used to reach this coloring includes, among other components, the search for a maximum clique.) Nevertheless, SSGC obtained here a better coloring (240) than GTS (242) and outperformed the previous best result (241) for this graph (Morgenstern, 1996). Our scatter search approach also improves in quality on the results obtained with tabu search (GTS) on a few graphs. This means that tabu search, the *improvement method* we used within scatter search, surely benefits from the other general components of scatter search.

4.4 Capacitated Multicommodity Network Design

The fixed-charge capacitated multicommodity network design formulation (CMND) represents a generic model for a wide range of applications in planning the construction, development, improvement, and operations of transportation, logistics, telecommunication, and production systems, as well as in many other major areas. The problem is usually modeled as a combinatorial optimization problem and is NP-hard in the strong sense. Thus, not only the generation of optimal solutions to large problem instances constitutes a significant challenge, but even identifying efficiently good feasible solutions has proved a formidable task not entirely mastered.

The goal of a CMND formulation is to find the optimal configuration - the links to include in the final design - of a network of limited capacity to satisfy the demand of transportation of different commodities sharing the network. The objective is to minimize the total system cost, computed as the sum of the link fixed and routing costs.

The paper Ghamlouche, Crainic and Gendreau (2001) proposed a new class of cycle-based neighborhood structures for the CMND and evaluated the approach within a very simple tabu-based local search procedure that currently appears as the best approximate solution method for the CMND in terms of robust performance, solution quality, and computing efficiency. Still more recently, Ghamlouche, Crainic and

Gendreau (2002) explore the adaptation of path relinking to the CMND. This work evaluates the benefits of combining the cycle-based neighborhood structures and the path relinking framework into a better meta-heuristic for this difficult problem.

The method proceeds with a sequence of cycle-based tabu search phases that investigate each visited solution and add elite ones to the reference set ***R***. When a predefined number of consecutive moves without improvement is observed, the method switches to a path relinking phase.

What solutions are included in the reference set, how good and how diversified they are, has a major impact on the quality of the new solutions generated by the path relinking method. We study six strategies corresponding to different ways to build ***R***.

In strategy ***S1***, ***R*** is built using each solution that, at some stage of the tabu search phase, improves the best overall solution and become the best one.

- Strategy ***S2*** retains the *"best" local minima* found during the tabu search phase. This strategy is motivated by the idea that local minimum solutions share characteristics with optimum solutions.

- Strategy ***S3*** selects *R-improving* local minima, that is local minimum solutions that offer a better evaluation of the objective function than those already in ***R.***

- Strategy ***S4*** allows solutions to be retained in ***R*** not only according to an attractive solution value but also according to a *diversity*, or *dissimilarity* criterion.

- Strategy ***S5*** aims to ensure both the *quality* and the *diversity* of solutions in ***R***. Starting with a large set of "good" solutions, ***R*** is partially filled with the best solutions found, to satisfy the purpose of quality. It is then extended with solutions that change significantly the structure of the solutions already in ***R*** to ensure diversity.

- Strategy ***S6*** proceeds similarly to ***S5*** with the difference that ***R*** is extended with solutions *close* to those already in ***R***.

During the path relinking phase, moves from the initial to a neighboring one direct the search towards the guiding solution. Due to the nature of the neighborhoods used, there is no guarantee that the guiding solution will be reached. One cannot, therefore, stop the process only if the current and the guiding solutions are the same. We then define Δ_{IG} as the number of arcs with different status between the initial and the guiding solutions and we allow the search to explore a number of solutions not larger than Δ_{IG}.

Initial and guiding solutions are chosen from the reference set. This choice is also critical to the quality of the new solutions and, thus, the performance of the procedure. We investigate the effect of the following criteria:

- ***C1***: Guiding and initial solutions are defined as the *best* and *worst* solutions, respectively.

- ***C2***: Guiding solution is defined as the *best* solution in the reference set, while the initial solution is the *second best* one.

- ***C3***: Guiding solution is defined as the *best* solution in the reference set, while the initial solution is defined as the solution with *maximum Hamming distance* from the guiding solution.

- ***C4***: Guiding and initial solutions are chosen *randomly* from the reference set.

- ***C5***: Guiding and initial solutions are chosen as the *most distant* solutions in the reference set.

- ***C6***: Guiding and initial solutions are defined respectively as the *worst* and the *best* solutions in the reference set.

The path relinking phase stops when the reference set becomes empty (cardinality ≤ 1). Then, either stopping conditions are verified, or the procedure is repeated to build a new reference set.

Extensive computational experiments, conducted on one of the 400 MHz processors of a Sun Enterprise 10000, indicate that the path relinking procedure offers excellent results. It systematically outperforms the cycle-based tabu search method in both solution quality and computational effort. On average, for 159 problems path relinking obtains a gap of 2.91% from the best solutions found by branch-and-bound versus a gap of 3.69% for the cycle-based tabu search. (However, the branch and bound code, CPLEX 6.5, was allowed to run for 10 CPU hours.) Thus, path relinking offers the best current meta-heuristic for the CMND.

4.5 A Scatter Search for the Maximum Clique Problem

The Maximum Clique Problem (MCP) can be defined as follows. Given an undirected graph G=(V, A) and A(v_i) denoting the set of vertices v_j such that (v_i, v_j) $\in$A, then a graph G1=(V1, A1) is called a subgraph of G if V1$\subseteq$V, and for every v_j $\in$V1, A1(v_i)=A(v_i)$\cap$V1. A graph G1 is said to be complete if there is an arc for each pair of vertices. A complete subgraph is also called a clique. A clique is maximal, if it is not contained in any other clique. In the MCP the objective is to find a complete subgraph of largest cardinality in a graph. The clique number is equal to the cardinality of the largest clique of G.

The MCP is an important problem in combinatorial optimization with many applications which include: market analysis, project selection, and signal transmission. The interest for this problem led to the algorithm thread challenge on experimental analysis and algorithm performance promoted by Second DIMACS Implementation Challenge (Johnson and Trick 1996).

Cavique, Rego and Themido (2001) developed an experimental study for solving the Maximum Clique Problem (MCP) using a Scatter Search framework. The proposed algorithm considers structured solution combinations weighted by a "filtering vector" playing the role of linear combinations. For the heuristic improvement a simple tabu search procedure based on appropriate neighborhood structures is used. Some special features and techniques have been introduced for the implementation to this problem. The algorithm implementation is structured into five basic methods:

Diversification Generation Method

The aim of the diversification generation method is to create a set of solutions as scattered as possible within the solution space while also using as many variables (or solution attributes) as possible. In the MCP, all vertices in the graph G should be present in RS.

When the algorithm starts, RS is initialized with a set of diverse solutions obtained by a constructive procedure, which starting from a single vertex, each step adds a new vertex to the current clique until a maximal clique is found. Starting from a different vertex not yet included in RS, the procedure is repeated as many times as the cardinality of the reference set. The clique value is used to order the solutions in RS.

Improvement Method

The improvement method has two phases: given a solution that is typically infeasible, the method first undertakes to recover to a feasible one; and afterward it attempts to increase the objective function value. Neighborhood structures based on add, drop, and node swap moves are used in the local search. The method allows for the solutions being infeasible by temporarily dealing with non complete subgraphs, which implements a strategic oscillation allowing trajectories to cross infeasible regions of solutions.

Reference Set Update Method

The reference set update method must be carefully set up with diverse and high quality solutions to avoid the phenomenon of *premature convergence* of RS, which occurs when all the solutions are similar. To prevent this "pitfall", the reference set RS is divided into two groups: the set of best solutions and the set of diverse solutions.

Regarding the replacement policy, a combination of the best replacement policy and the worst replacement policy called 'journal replacement' policy is used, which replaces the worst solution with the new best solution found, reporting all the 'hits' of the search.

Subset Generation Method

This method generates the following types of solution subsets which are combined in the next method. The method generates subsets with two, three or more elements in a relatively reduced computational effort. To eliminate repetition of the elements in the subsets, the reference set with diverse solutions is used for the two by two combinations, instead of the complete reference set. The method also includes a new feature by adding a distant (or diverse) solution maximizing the distance from the region defined as the union of the vertices in the solution's subset. In this way, a new point "far from" the solution cluster is obtained at each iteration to maintain an appropriate diversity of solutions in the reference set.

Solution Combination Method

This method uses each subset generated in the subset generation method and combines the subset solutions, returning one or more trial solutions. Solution combinations are created using a filter vector applied to the union of solutions, called λ-filter. The λ-filters are used in Scatter Search as a form of structured combinations of solutions. Instead of drifting within the solution space defined by the reference set, the SS procedure searches each region extensively by applying different λ-filters. Each λ-filter generates a trial solution to be improved by the Improvement Method. A sequence of previously planned λ-filters generates a set of solutions within and beyond regions defined by two or more solutions in which new trial solutions will be chosen for updating the reference set in an evolutionary fashion.

Applying λ-filters in subsets with diverse solutions, a bypass strategy is created. Instead of finding solutions between two others, it is possible to bypass the path using a intermediate reference solution.

Computational results obtained on a set of the most challenging clique DIMACS benchmark instances shown the scatter search algorithm can be advantageously compared with some of the most competitive algorithms for the MCP.

4.6 Assigning Proctor to Exams with Scatter Search

Several real assignment problems can be viewed as a generalization of the well-known Generalized Assignment Problem. One of these problems is the Proctor Assignment Problem (PAP), which consists in the assignment of proctors to final exams at a school or university, with respect to some objective function as for example the maximization of the total preferences of proctors to exams' dates.

Martí, Lourenço and Laguna (2000) presented a Scatter Search to solve particular instances of the PAP, based on the real data from a Spanish University. The problem was formulated as a multiobjective integer program with a total preference and workload-fairness objective functions, and can be stated as follows: consider a set of proctors at a large university. Each proctor has a maximum number of hours that he/she can devote to proctor final exams. This limit depends on his/her contract and teaching load. Each final exam requires a given number of proctors for proctoring. Since the most of the proctors are graduate students and Teaching Assistants (TAs), they also have final exams and therefore they cannot proctor exams during periods that conflict with their own exams. The constraints can be summarized as follows:

- Each exam must be proctored by a specified number of TAs.
- A TA cannot exceed his/her maximum number of proctor hours.
- A TA cannot proctor more than one exam at the same time.
- A TA cannot proctor a final exam that conflicts with one of his/her own.
- A TA should proctor the exams of the courses he/she taught.

The last constraint can be handled before formulating the model by simply assigning proctors to the exams of the courses they taught and adjusting the associated input data accordingly (e.g., reducing the total number of proctor hours and the exam requirements). Teaching assistants have preferences for some exams, which reflect their desire for proctoring on a given day or avoiding certain days. For example, some TAs would like to avoid proctoring an exam the day before one of their own exams. As a result of these preferences, one objective of the problem is to make assignments that maximize a function of the total preferences.

Another important criterion is the assignment of proctor to exams such that the workload is evenly distributed among TA's. Unfair workloads are likely to generate conflicts among TA's and between TA's and the administration. Several objective functions can be formulated to measure the workload-fairness of a given assignment. One possibility is to maximize the minimum workload associated with each TA. Since the number of available hours for each TA varies, the workload can be expressed as the ratio of assigned hours to available hours.

Martí, Lourenço and Laguna (2000) considered a weighted function to deal with the multiobjective model and proposed a scatter search method based on the work by Glover (1998), Laguna (1999) and Campos et al. (1998). The diversification generation method generates the population solutions using the preferences values modified by a frequency function. This frequency function is used to bias the potential assignment of TAs to exams during subsequent constructions of solutions, and therefore to induce diversity in the new solutions with respect to the solutions already in the population. TAs are assigned to exams in order to maximize the modified preference values. The reference set is constructed by using the best solutions and a distance function between solutions to diversify the solutions in this set. The solution combination method is applied to each subset generated as in Glover (1998). It is based on a voting system, where each solution votes for specific assignment of TAs to exams. The resulting solution may be infeasible with respect to some constraints, and in this case, a repair mechanism is applied. The method outputs the best solution with respect to the weighting function.

The data used for these experiments correspond to real instances of the proctor assignment problem at the Universitat Pompeu Fabra in Barcelona (Spain). The results were compared with manually generated assignments also with assignments found by solving the mixed-integer programming formulation with Cplex 6.5 (some of which are optimal). For the set of test problems that utilize the utility function, the scatter search solutions are often slightly sub-optimal. However, this is offset by the advantage that the scatter search reference set contains a number of high-quality solutions, allowing the decision-maker to choose the one to implement, based on non-quantitative elements. The maximum standard deviation of the utility function value for solutions in the final reference set was 0.000407 for all problem instances. This indicates that practically all of the solutions in the final reference set have the same quality with respect to the objective function value. Since the utility function is a mathematical representation of some subjective measure of performance associated with a given assignment, the ability to choose among solutions that have similar objective function values is an important feature of a decision support system designed for this managerial situation.

Since scatter-search is a population-based search, the method is a useful solution technique to solve multiobjective problems by finding an approximation of the set of Pareto-optimal solutions. A multiobjective scatter search for the solving the PAP is investigated in Lourenço, Martí and Laguna (2001). The main features of this approach are the construction and updating of the reference set using the set of non-dominated solutions. Also, the cardinality of the reference set varies with respect to the size of the set of non-dominated solutions. An improvement method is applied to improve the solutions obtained by the greedy heuristic, the diversification method and the solution combination method. This improvement method consists of a simple local search method, where the neighborhood is obtained by exchanging one TA for another one from the list of proctors. Finally, the method is restarted with the set of non-dominated solutions in the reference set. Preliminary results for this new approach indicate that multiobjective scatter search with restarting gives the best results with respect to the weighting function, across different versions of the method. Also, multiobjective scatter search enables the user to analyze a collection of very good solutions and make the final decision.

4.7 Periodic Vehicle Loading

Delgado, Laguna and Pacheco (2002) address a logistical problem of a manufacturer of auto parts in the north of Spain. The manufacturer stores auto parts in its warehouse until customers retrieve them. The customers and the manufacturer agree upon an order pickup frequency. The problem is to find the best pickup schedule, which consists of the days and times during the day that each customer is expected to retrieve his/her order. For a given planning horizon, the optimization problem is to minimize the labor requirements to load the vehicles that the customers use to pick up their orders.

Heuristically, the authors approach this situation as a decision problem in two levels. In the first level, customers are assigned to a calendar, consisting of a set of days with the required frequency during the planning horizon. Then, for each day, the decision at the second level is to assign each customer to a time slot. The busiest time slot determines the labor requirement for a given day. Therefore, once customers have been assigned to particular days in the planning horizon, the second-level decision problem is equivalent to a multiprocessor scheduling problem (MSP), where each time slot is the equivalent of a processor, and wherethe objective is to minimize the makespan.

A scatter search procedure is developed for the problem of minimizing labor requirements in this periodic vehicle-loading problem and artificial as well as real data are used to assess its performance. The scatter search constructs and combines

calendar assignments and uses a heuristic to solve the MSP's for each day in the planning horizon and thus obtain a complete solution.

The diversification method is based on GRASP constructions. The greedy function calculates the increase in labor requirements from assigning a previously unassigned order to a calendar. The procedure starts with all the orders in the "unassigned" set. The orders are considered one by one, from the largest to the smallest (i.e., from the one that requires the most amount of labor to the one that requires the least amount of labor).

The improvement method is based on a procedure that changes the assignment of an order from its current calendar to another. Preliminary experiments showed that the performance of the improving method with simple moves (i.e., the change of calendars for one order only) was not as good as the performance of a local search employing composite moves. A composite move is a chain of simple moves. Therefore, while a simple move prescribes the change of one order from one calendar to another, a composite move prescribes the change of several orders from their current calendars to others. It may seem that a local search based on simple moves should be capable of finding sequences of moves that are equivalent to composite moves. However, this is not necessarily the case because the local search based on simple moves is greedy and searches for the best exchange and performs the exchange only if it results in an improving move. A local search with composite moves, on the other hand, may perform some non-improving simple moves that lead to a large improving move.

The combination method generates new solutions by combining the calendar assignments of two reference solutions. The objective function values of the reference solutions being combined are used to probabilistically assign orders to calendars in the new trial solution. That is, on the average, most of the assignments come from the reference solution with the better objective function value. The procedure uses a static update of the reference set.

Using both randomly generated data adapted from the literature and real data from a manufacturer, the authors were able to show the merit of the scatter search design. In particular, extensive experiments show that significant savings may be realized when replacing the manufacturer's current rules of thumb with the proposed procedure for planning purposes.

4.8 Tabu and Scatter Search in Job-Shop Scheduling

The job-shop scheduling problem is known as a particularly hard combinatorial optimization case. It arises from OR practice, has a relatively simple formulation, excellent industrial applications, a finite but potentially astronomical number of solutions and unfortunately is strongly NP-hard. It is also considered an indicator of practical efficiency of advanced scheduling algorithms. In the early nineties, after a series of works dealing with optimization algorithms of the B&B type, it became clear that pure optimization methods for this problem had a ceiling on their performance. In spite of important advances over the past two decades, the best B&B methods cannot solve instances with more than 200 operations in a reasonable time (hours, days, weeks).

A new era started when job-shop algorithms based on the TS approach appeared. The simple and almost ascetic Algorithm TSAB (Nowicki and Smutnicki , 1996), designed originally in 1993, found the optimal solution of the notorious job-shop instance FT10 (100 operations) in a few seconds on a PC. This instance had waited 26 years, since 1963, to be solved by an optimization algorithm. But going far beyond the solution of FT10, the TSAB approach made it possible to solve, in a very short time on a PC, instances of size up to 2,000 operations with unprecedented accuracy – producing a deviation from an optimality bound of less than 4% on average. This is considerably better than the deviation of approximately 20% for special insertions technique, 35%

for standard priority rules and over 130% for random solutions. Another highly effective tabu search method for the job shop problem has recently been introduced by Grabowski and Wodecki (2001).

Further exploration of the ideas underlying TSAB focuses on two independent subjects: (1) acceleration of the speed of the algorithm or some its components, and (2) a more sophisticated diversification mechanism, the key for advanced search scattering. Recent papers by Nowicki and Smutnicki (2001a and 2001b), provide some original proposals located precisely in these research streams. They refer to a new look at the landscape and valleys in the solution space, set against the background of theoretical properties of various distance measures. There are proposed *accelerators* based on theoretical properties, which, by means of skillful decomposition and aggregation of calculations, speed up significantly search process, namely: (a) INSA accelerator (advanced implementation of insertion algorithm used for starting solution in TSAB), (b) tabu status accelerator, (c) NSP accelerator (fast single neighborhood search). Next, in order to diversify the search, TSAB has been embedded in the Scatter Search and Path Relinking framework. The resulting algorithm ***i***-TSAB described in Nowicki and Smutnicki (2001a), the powerful successor of TSAB, works with elite centers of local search areas forming a MILESTONE structure, modified by space explorations conducted from VIEWPOINTS located on GOPS (a class of goal oriented paths).

As the immediate practical result of this new approach, 24 better upper bounds (new best solutions) have been found for 24 of the 35 instances from the common benchmark set of Taillard, attacked by all job-shop algorithms designed till now. The proposed algorithm still runs on a standard PC in a time of minutes.

4.9 A Tabu Scatter Search Metaheuristic for the Arc Routing Problem

The problem treated in Greistorfer (2001a) is the so-called *capacitated Chinese postman problem* (CCPP). The goal of the (undirected) CCPP is to determine a least-cost schedule of routes in an undirected network under the restriction of a given fleet of vehicles with identical capacity, which operates from a single depot node. In the standard version of the CCPP the number of vehicles is unlimited, i.e. it is a decision variable. The CCPP is a special instance of the general class of arc routing problems, a group of routing problems where the demand is located on arcs or edges (one-way or two-way roads) connecting a pair of nodes (junctions). Relevant practical examples of the CCPP are postal mail delivery, school bus routing, road cleaning, winter gritting or household refuse collection. But applications are not limited to the routing of creatures or goods. There are also cases in industrial manufacturing, e.g. the routing of automatic machines that put conducting layers or components on to a printed circuit board.

The algorithmic backbone of the *tabu scatter search* (TSS) metaheuristic introduced is a tabu search (TS) which operates with a set of neighborhood operators (edge exchange and insert moves) on a long-term diversification strategy guided by frequency counts. The short-term tabu memory works with edges and simply prohibits reversal moves within a dynamically varied tenure period. Additionally, the procedure has a pool component which accompanies the TS by maintaining a set of elite solutions found in the course of the optimization. If the classic genetic algorithm can be understood as a pure parallel pool method because it always works on a set of high quality solutions, then the TSS follows a sequential pool design, where periods of isolated and single-solution improvements of the TS alternate with multi-solution combinations. With respect to the type of encoding, the solution combination method (SCM) is purely phenotypical, which turns this pool method into a type of scatter search (SS) algorithm. The TSS architecture as proposed here does not exactly follow the template ideas of Glover (1998), although there are many common features as will be outlined below.

The SCM component combines elite solutions which have been collected by the TS. As suggested in the template paper and further literature, the combination of solutions is supported by generalized rounding procedures which may follow heuristic rules or approaches based on linear programming (LP). The underlying principle of the SCM proposed follows an adapted transportation problem (TPP) formulation, which is the generalization of the assignment operator of Cung et al. (1997).

The TSS and its SCM benefit from a data structure which, generally, can be used for problems described by pure permutations (customers) or by permutations where sub-strings (routes) have to be considered as well. The SCM works as follows. Given a set of elite solutions $S_1,...,S_c$, the TPP coefficients a_{ij} denote the number of times a customer j is assigned to a route i in this pool subset. The TPP coefficient matrix can be interpreted as an *assignment frequency matrix (AFM)*, being the linear combination of the c individual assignment matrices. Unit TPP demands are a consequence of the need that every edge has to be serviced by a single vehicle. The supply of a route is approximated by the average number of customers that can be serviced respecting the vehicle capacity whereas a dummy column picks up the oversupply. Maximizing this TPP results in customer-route assignments which maximize the total number of desirable assignments while simultaneously minimizing the total Euclidean distance to the *AFM,* which represents the (infeasible) combination of the initial trial points. Although the outcome of this SCM can be directly used, it is clearly improvable since the optimal clusters (sets) provided do not imply any guidance of how the vehicle routes (sequences) should be formed. Therefore, a *greedy sequencing heuristic* (GSH) is used for post-optimization to put the customers of all routes into a cost-convenient order.

The overall TSS starts from a random pool whose elements are exchanged for solutions which have been improved in the TS phase. The SCM is occasionally called and forms a combined solution which is then returned to the TS for further inspections. This alternating process between TS and the SCM stops after a pre-defined period of iterations.

The TSS is tested on several classes of CCPP instances: there are a number of planar Euclidean grid-graph instances, Euclidean random instances and the well-known DeArmon data set as used for the CARPET arc routing heuristic of Hertz et al. (2000). In a direct comparison with an old TS method (see Greistorfer (1995)) the TSS significantly improves the results for the Euclidean classes (in 54% of all cases) and is clearly able to keep up with CARPET regarding the instances from literature. Here the number of (known) optimal solutions is identical, while the TSS finds one more best-known solution. Its worst average deviation (due to a single instance) is only 1.29% higher than the one of CARPET. The total running times, which are scaled with respect to the CARPET-PC, are longer. However, it is shown that on average the TSS obtains its best results faster than the CARPET heuristic. Thus, adding a pool component to a TS and using an advanced SCM has obvious merits.

Testing Population Designs

The theme of Greistorfer (2001a) is continued in Greistorfer (2001b), where the focus is more on the methodology. The task is to work out relevant pool strategies and to evaluate them by means of thorough computational comparisons. Test results again refer to a sample of CCPP arc routing instances but, as mentioned above, the encoding offers a certain ability to generalize the algorithmic findings for a number of different problems. From the manifold design options for heuristic pool methods, the discussion concentrates on three basic components: the *input* and *output* functions, which are responsible for pool maintenance and which determine the transfer of elite solutions, and a solution *combination* method which must effectively combine a set of elite solutions provided by the output function.

The heuristic design variants of the TSS comprise four input strategies, $I_{0,\ldots,3}$, four output strategies, $O_{0,\ldots,3}$, and three SCMs, namely $M_{0,\ldots,2}$ (including the settings of Greistorfer (2001a), indexed with a *0)*.

Input strategy I_0 reduces the quality aspect to the cost dimension of a solution and does not deal with structural properties of a solution or their relations to each other. I_1 overcomes this disadvantage by including full duplication checks between potential elite solutions and pool members. I_2, known as the *reference set update method* of Glover (1998), is additionally linked with preceding hash comparisons. In I_3 an attempt is made to find a compromise by skipping the full duplication part in I_2, i.e. I_3 only relies on hashing. Input functions $I_{1,2,3}$ build on ordered structures which are provided by a *sorting algorithm*. A corresponding procedure is suggested by the author as well as a relevant *hash function*.

By analogy to I_0, O_0 does not utilize structural information and simply refers to a random selection. O_1 uses frequency counts and selects those solutions for combinations which have not been used before or have been used rarely. Such a tracking of the number of involvements of a solution in a combination process introduces a certain memory effect. By contrast, the *subset generation method* of Glover (1998) explicitly makes use of an algorithmic structure which completely avoids a duplicate subset selection. Output strategies O_2 and O_3 select solutions which have the smallest and largest distance to each other, respectively. For that purpose a *distance function* is proposed which aggregates the customer positions and their route membership.

The combination strategy M_0 is the LP-based transportation method which is described in detail in Greistorfer (2001a). The last two SCM variants are based on deriving average solutions. M_1 constructs average customer labels whereas M_2 determines average customer positions (see also Campos et al. (1999)). Both approaches relax the capacity restriction whose validity has to be secured afterwards by splitting the permutation sequence into a set of route clusters. The final offspring is obtained after applying the post-optimizing GSH.

The computational investigation of the results for the different TSS designs was performed by checking all possible $4 \cdot 4 \cdot 3 = 48$ TSS configurations against each other. Each configuration was run over the whole set of test instances and evaluated by its corresponding average objective function value derived from the best solutions found. The best configuration turned out to be (I_3,O_3,M_0). In order to evaluate the specific effects of an input, output or combination variant, all results were checked according to a *variance analysis* by means of SPSS.

Generally, effects of variations tended to be smaller at the input side of the pool since all tests did not indicate any significant difference between the input strategies described. One model of explanation is that duplications are effectively prevented by each of the input procedures $I_{1,2,3}$, while the (empirical) probability of collision (hashing error) is smaller than 0.195% for $I_{2,3}$. Another reason might be that strongly diversifying input strategies are not adequately utilized by the other variable components. The results generated by the straightforward algorithmic setting of I_0 cannot keep up with the results of the other methods.

The picture completely changes when output strategies or different SCMs are looked upon. It was found that the min-distance approach in O_2 is definitely an inferior option. The argument that good solutions are mostly found in the vicinity of the best solution cannot be upheld, which is clearly in agreement with the SS philosophy of selecting diverse solutions to be combined. This fact is also underlined by the superiority of the max-distance function O_3 over O_2. The expected memory effect in O_1 turned out to be too small to guide the selection process. Random sampling in O_0 can be justified in an isolated view which ignores input and combination method effects.

As an SCM, the LP approach of M_0 significantly contributes to finding better solutions than M_1 and M_2. While there are no significant relations between the latter ones, the individual best choice for M_0 proves its very useful role in the collective optimal design (I_3,O_3,M_0).

4.10 A Population Based Approach to the Resource Constrained Project Scheduling

Valls et al. (2001) propose a population-based approach to the resource-constrained project-scheduling problem (RCPSP), where n activities have to de processed, taking into account the precedence relations and the resource restrictions. The procedure incorporates different strategies for generating and improving a population of schedules. The method has two phases. Phase 1 (Figure) can be interpreted in terms of the scatter search methodology.

Figure. Phase_1 outline

1. POP = INITIAL_SET_1(20)
2. POP' = HIA(POP)
3. For [i=1,2]
  3.1. POP5 = {the 5 best schedules in POP'}
  3.2. POP = CSA(4,POP5)
  3.3. POP40 = {the 40 best schedules in POP}
  3.4. POP' = HIA(POP40)
4. Return the best schedule obtained

INITIAL_SET_1(*size*) is the Diversification Generation Method, which generates a collection of diverse trial solutions. To achieve quality and diversity, different well-known priority rules and random procedures are used. The best *size* solutions are stored in the set POP.

The Improvement Method is called HIA, the Homogenous Interval Algorithm. It is applied to the solutions in POP and is an iterative procedure for improving the local use of resources. It incorporates an oscillatory mechanism that alternatively searches two different regions of the schedule space (strategic oscillation).

The Subset Generation Method and the Solution Combination Method are carried out by the Convex Search Algorithm (CSA). CSA(k,SET) generates all pairs of reference solutions of SET and combine each of them with a procedure that integrates path relinking characteristics.

First of all, CSA codifies each schedule by a topological order (TO) representation. A TO representation of a schedule S is a special priority value vector γ, the one that fulfills the conditions (1) $\{\gamma(i), i = 1, ..., n\} = \{1, ..., n\}$, (2) $s_i < s_j \rightarrow \gamma(i) < \gamma(j)$, being s_i the beginning of activity i in S, and (3) $s_i = s_j$ and $i < j \rightarrow \gamma(i) < \gamma(j)$, i.e., the label is used to order activities with the same beginning. The Serial schedule generation scheme can be used to decodify a TO representation γ and obtain an active schedule $S(\gamma)$, by selecting at each stage the eligible activity j with the lowest priority $\gamma(j)$.

Afterwards, for each pair of reference schedules A and B, CSA calculates the priority value vectors γ^p, defined by $\gamma^p(j) = (1-p/2^k)\ \gamma_A(j) + (p/2^k)\ \gamma_B(j)$, p = 1, ..., 2^k-1, where k is an integer and γ_A and γ_B are the TO representations of A and B, respectively. Although γ^p is not a TO representation, is a vector of priorities compatible with the precedence relations that can be easily transformed into a TO representation that gives the same schedule as γ^p when decodified. Moreover, the priority vectors γ^p, p = 1, ..., 2^k-1 are uniformly distributed in the geometric segment joining γ_A and γ_B and it can be statistically proved that something similar happens to the schedules $S(\gamma^p)$, p = 1, ..., 2^k-1, in the path between $S(\gamma_A)$ and $S(\gamma_B)$. This means that each $S(\gamma^p)$ incorporates more

attributes of $S(\gamma_B)$ and less of $S(\gamma_A)$ as p increases, so this procedure builds a trajectory between $S(\gamma_A)$ and $S(\gamma_B)$.

The best 40 schedules calculated by CSA are introduced in POP40, and the improvement procedure HIA is applied to each of them. The Reference Set Update Method is quite simple: the first reference set contains the best 5 schedules calculated in step 2 and the following reference sets are formed by the best 5 solutions obtained after having applied CSA and HIA. So, the reference set is totally replaced in each iteration, looking for a fast convergence, since step 3 is applied only twice.

At the end of phase 1, the best solution obtained so far is generally of high quality. Experience seems to indicate that good candidate schedules are usually to be found 'fairly close' to other good schedules. The objective of the second phase is therefore to closely explore regions near high quality sequences. Exploring such a region means, first, generating a population by taking a random sample from a region near a good sequence, and second, applying to the population the improving procedure used in the first phase – but with a variation. The on-going search is interrupted when a better sequence is obtained and a fresh search starts from the point of the newer sequence. Phase two begins from the best solution obtained in phase one.

Computational experiments have been carried out on the standard j120 set generated using ProGen. They show that the indicated SS algorithm produces higher quality solutions than state-of-the-art heuristics for the RCPSP in an average time of less than five seconds in a PC at 400 MHz.

4.11 Multiple Criteria Scatter Search

Beausoleil (2001) has developed a scatter search procedure that uses the concept of Pareto optimality to obtain a good approximate Pareto frontier. Tabu Search is used to obtain an initial set of reference points. Different frequency memories are used to diversify the search. In order to designate a subset of strategies to generate a reference solutions a choice function called Kramer Selection is used. A Kemen-Snell measure is applied in order to find a diverse set to complement the subset of high quality current Pareto solutions. Path Relinking and Extrapolated Path Relinking are used as a Combination Method.

Structured weighted combination is used in a special case to obtain weighted solutions inside the convex region spanned by selected reference points. To implement the process, in the Tabu Search phase, memory is maintained of selected attributes of recent moves and their associated solutions. A thresholding aspiration guides the selection of an initial set of solutions. Solution quality is measured by introducing an Additive Value Function in this phase. A study involving multiple cases demonstrates the ability of the algorithm to find a diverse Pareto frontier.

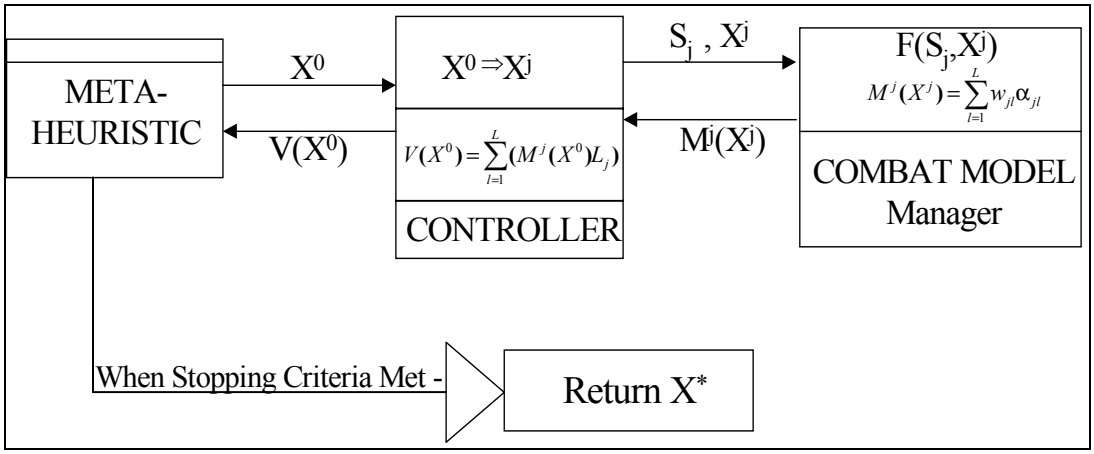
The results of the experiments show that the first TS phase generates an initial good Pareto frontier. The combined method using path relinking and extrapolated path relinking as an intensification-diversification method proves an effective mechanism to generate new Pareto points, yielding a good approximation to the Pareto frontier in a relatively small number of iterations.

4.12 Meta-Heuristic Use of Scatter Search via OptQuest

Optimization and simulation models play an important role in Defense Analyses. A crucial component of a model-based analysis effort is agreement upon the planning scenario upon which the analysis is conducted. For example in military force structuring, the model might suggest a prescribed force structure necessary to best meet the demands of a planning scenario. Conversely, given some proposed force structure, a model might provide insight into a "best use" of that force within the specified scenario. A particularly perplexing challenge for military analysts occurs

when they must suggest a single overall force structure after considering multiple competing scenarios, each suggestive of potentially differing optimal force structures.

Hill and McIntyre (2000) addressed this particular vexing military force structure problem. They define a **robust force structure (solution)** as that force structure (solution) "providing the best overall outcome as evaluated with respect to some set of scenarios each of which has an associated likelihood of occurrence." Their approach considered the multi-scenario optimization problem within which each particular scenario solution becomes a component of an aggregate multi-scenario solution. They treat the multi-scenario space as a composite of the component scenario spaces where each component space contributes relative to its likelihood of occurring or relative importance weight. Using a meta-heuristic to guide a search using the combat model as an evaluation function provides a means to find a single solution, potentially optimal in the multi-scenario space, and by definition, robust across each of the individual scenario spaces.



The Hill and McIntyre approach is presented graphically in the Figure above. Central to the approach is a CONTROLLER interface between the meta-heuristic module and the combat models conducting the evaluations. The META-HEURISTIC guides the search process providing the CONTROLLER potential solutions (input force structure) and receiving from the CONTROLLER evaluations of those solutions. The COMBAT MODEL receives its input (and scenario) from the CONTROLLER, evaluates the input, and returns the requisite quality measure from the combat model assessment associated with the input. The CONTROLLER accepts the potential solutions, provides those to each of the scenario evaluators in the COMBAT MODEL, and combines each measure into the final value or fitness of the potential solution. This process continues until predefined stopping conditions are satisfied at which time the best, or set of best, solutions are returned.

Bulut (2001) applied scatter search, implemented within the OptQuest callable library (Laguna and Marti, 2002) to solve a multi-scenario optimization problem based on the United States Air Force's Combat Forces Assessment Model (CFAM), a large-scale linear programming model for weapons allocation analyses. Three notional planning scenarios were used and a robust solution sought to the multi-scenario problem. He compared OptQuest results with previous results obtained using a genetic algorithm (for the same scenarios). His results indicated that better overall solutions, a greater diversity of solutions, and quicker convergence results were obtained using the OptQuest scatter search approach.

The methodology proposed by Hill and McIntyre (2000) and implemented by Bulut (2001) using OptQuest is directly applicable to any analytical situation involving

competing "scenarios" within which one needs a single solution that is "robust" relative to any differences among the scenarios.

4.13 Pivot Based Search integrated with Branch and Bound for Binary MIPs

Linear programming models with a mixture of real-valued and binary variables are often appropriate in strategic planning, production planning with significant setup times, personnel scheduling and a host of other applications. The abstract formulation for linear problems with binary integers takes as data a row vector **c**, of length n, a $m \times n$ matrix $\boldsymbol{A}$ and a column vector $\boldsymbol{b}$ of length m. Let D be the index set $1,\ldots,n$. The problem is to select a column vector, $\boldsymbol{x}$ of length n so as to

$$\min \sum_{i \in I} c_i x_i$$

s.t.

$$Ax \geq b$$

$$x_i \in \{0,1\} \qquad i \in I$$

$$x_i \geq 0 \qquad i \in D \setminus I$$

where the index set I gives the variables that must take on zero-one values.

Issues related to the behavior of a pivot based tabu search integrated with branch and bound algorithm, using path relinking and chunking are discussed by Løkketangen and Woodruff (2000).

The integration takes place primarily in the form of local searches launched from the nodes of the branch and bound tree. These searches are terminated when an integer feasible solution is found or after some number of pivots, *NI*. Any time a new best is found, the search is continued for an additional *NI* pivots. Chunking (see Woodruff 1996, 1998) is used to detect solutions that should be used for special path relinking searches that begin at the LP relaxation and to determine when the use of pivot searches should be discontinued. (See also Glover, Løkketangen and Woodruff, 2000, for another application of chunking to the same kind of problems)

As the search is launched from nodes in a B&B tree, there are some special considerations that come into play that sets this use of the pivot based search somewhat apart from other implementations. First, the chunking mechanism and the path relinking based target searches, respectively, fulfill the functions of diversification or intensification. Second, the purpose, or focus, of the search is somewhat different from the stand-alone search, in that for some of the searches, the emphasis is shifted more towards obtaining integer feasibility quickly. This focus is controlled by a separate parameter *skew*, that is used to adjust the relative importance of obtaining feasibility versus maintaining a good objective function value.

Chunking addresses the questions of when the launching of pivot based searches should be terminated, and when the path relinking searches should be launched. More specifically, path relinking searches are used to exploit "unique" or "outlying" solutions. The meaning of "unique" and "outlying" can be supplied by chunking.

Two types of local searches can be launched at a node. The first are the normal TS pivot-based searches launched from nodes in the B&B tree (see Løkketangen and Glover, 1995, 1996, 1998, 1999).

The other type are the Path Relinking searches. After a best-so-far solution $\boldsymbol{x}^*$ has been found, the chunking mechanisms try to identify distant solutions, **x'** (w.r.t. the current sample). When such a distant solution has been identified, a *2-target* search is launched. This is a variant of path relinking, with the purpose of launching a new

search into unknown territory, while at the same time keeping good parts of the solutions. This integrates the considerations of intensification and diversification.

The starting point of this search is the relaxed root node solution LP^* (being an upper bound on the objective function value), and the target for the path relinking is the hyperplane defined by the common integer solution values of $\boldsymbol{x}^*$ and **x'**. All integer variables are freed. To allow the search to focus on this hyperplane, the integer infeasibility part of the move evaluation function is temporarily modified. (The objective function value component is unaltered, as is the aspiration criterion – see Løkketangen and Woodruff, 2000). Instead of using the normal integer infeasibility measure of summing up over all the integer variables the distance to the nearest integer, the authors use the following scheme:

- Sum up over all the integer variables
- If the two targets have the same integer solution value for the variable, use the distance to this value.
- If the two targets differ, use the normal integer infeasibility measure (i.e. the closest integer value).

When the search reaches the hyperplane connecting $\boldsymbol{x}^*$ and **x'**, the normal move evaluation function is reinstated, and the search continues in normal fashion for *NI* iterations.

Computational testing was done on problems from Miplib and Dash Associates, consisting of a mix of MIP's and IP's. The testing showed that the local searches had a beneficial effect on the overall search time for a number of problem instances, particularly those that were harder to solve.

4.14 Scatter Search to Generate Diverse MIP Solutions

Often, scatter search and star path algorithms (Glover 1995), generate diverse sets of solutions as a means to an end. In a recent paper by Glover, Løkketangen and Woodruff (2000) diversity is the ultimate goal for which scatter search and star paths are employed. This paper presents methods of systematically uncovering a diverse set of solutions for 0-1 mixed integer programming problems. These methods can be applied to instances without any special foreknowledge concerning the characteristics of the instances, but the absence of such knowledge gives rise to a need for general methods to assess diversity.

When the objective function is only an approximation of the actual goals of the organization and its stakeholders, the one solution that optimizes it may be no more interesting than other solutions that provide good values. However, information overload can be a problem here as well. It is not desirable to swamp the decision maker with solutions. Highly preferable is to identify a set of solutions that are decently good and, especially, diverse. One can reasonably rely on the objective function to quantify the notion of "decently good". The diversification methods given by Glover, Løkketangen and Woodruff are based on the idea of generating extreme points in a polyhedral region of interest and then using these points and the paths between them in a variety of ways. The methods examine points on the polyhedron, within and "near" it. Their algorithm proceeds in two phases: first it generates a set of *centers* and then connects them using *star paths*.

The description of the generation of centers can also be broken into two phases. First a diversification generator is used to create points. In the second phase, these points are provided as data to an optimization problem that results in extreme points that are averaged to create the centers.

Although a diverse set of good solutions is clearly desirable, it is not clear in advance how to measure the property of diversity. In spite of the fact that the objective function

is not exact, it presumably gives a reasonable way to assess the relative "goodness" of a set of solutions. No such simple mapping is known from solution vectors to a one-dimensional measure of diversity. Diversity measures are required both for the design of practical software and for research purposes. For practical software, it is important to know if the user should be "bothered" with a particular solution vector – that is, to know if a vector adds enough diversity to warrant adding it to the set of solutions that are displayed. For research purposes, one might want to compare the set of vectors generated by one method with a set of vectors generated by another.

There are a number of advantages to the quadratic metric known in this context as *Mahalanobis* distances. This metric is scale invariant and can take correlations into account if based on a covariance matrix. Furthermore, this type of distance connects naturally with a scalar measure of the diversity of a set of vectors, which is the determinant of the covariance matrix of the set. Under the assumption of multivariate normality, the covariance matrix defines ellipsoids of constant Mahalanobis distances that constitute probability contours. Large covariance determinants correspond to large volumes in the ellipsoids. The assumption of multivariate normality is not needed to use the covariance determinant to put an order on sets of vectors and furthermore it is not needed to see that adding points with large Mahalanobis distances will increase the covariance determinant.

However, there is a major difficulty. In order to calculate a covariance matrix for a set of vectors of length $p = n$ one must have a set of vectors that does not lie entirely in a subspace. This means that at a minimum the set must contain $n + 1$ vectors and for MIP solutions, more vectors will often be required to span the full n dimensions. For even modest sized MIPs this is not good. In order to have a working definition of diversity, one must have thousands of solution vectors. A remedy for this difficulty that also increases the plausibility of multivariate normality has been referred to as *chunking* by Woodruff (1998). A generalization based on principal components has also been proposed by Woodruff (2001).

As general purpose optimization methods are embedded in decision support systems, there will unquestionably be an increased need not only for optimal solutions, but also for a diverse set of good solutions. Scatter search and star paths can be an effective means to this end.

Results of computational experiments demonstrate the efficacy of the "scatter-star-path" method for generating good, diverse vectors for MIP problems. Furthermore, the results show that the method offers particular advantages when used in conjunction with brand and bound. The creation of these results illustrates the use of methods for measuring the diversity for a set of solutions.

4.15 Path Relinking to Improve Iterated Re-start Procedures

Research has been performed to investigate the ability of path relinking to improve the performance of iterated re-start procedures, with attention focused in particular on the GRASP method (Ribeiro and Resende, 2002). One possible shortcoming of the standard GRASP algorithm is the independence of its iterations, i.e., the fact that it does not learn from the history of solutions found in previous iterations. This is so because it discards information about any solution encountered that does not improve the incumbent. Information gathered from good solutions can be used to implement extensions based on path-relinking.

Path relinking was originally proposed in the context of tabu search as an intensification strategy which explores trajectories connecting high-quality solutions. The use of path relinking within a GRASP procedure was first proposed by Laguna and Martí (1999), being followed by several extensions, improvements, and successful applications (e.g., Canuto et al, 2001; Aiex et al, 2002; Ribeiro et al. 2002). Path relinking and a very short term memory used within the local search were

instrumental to make a recently proposed GRASP heuristic for the capacitated minimum spanning tree problem competitive with other approaches in the literature (Souza et al. 2002). Two basic strategies are used to apply path relinking in the context of a GRASP heuristic:

- apply path relinking as a post-optimization step to all pairs of elite solutions; and

- apply path relinking as an intensification strategy to each local optimum obtained after the local search phase

Both strategies maintain and handle a pool with a limited number Max_Elite of elite solutions found along the search (we used Max_Elite ranging from 10 to 20 in most implementations). The pool is originally empty. Each locally optimal solution obtained by local search is considered as a candidate to be inserted into the pool if it is sufficiently different from every other solution currently in the pool. If the pool already has Max_Elite solutions and the candidate is better than the worst of them, then the former replaces the latter. If the pool is not full, the candidate is simply inserted.

Applying path relinking as an intensification strategy to each local optimum seems to be more effective than simply using it as a post-optimization step. In this context, path relinking is applied to pairs $X - Y$ of solutions, where X is the locally optimal solution obtained after local search and Y is one of a few elite solutions randomly chosen from the pool (usually only one elite solution is selected). The algorithm starts by computing the symmetric difference between X and Y, resulting in a set Δ of moves which should be applied to one of them (the initial solution) to reach the other (the guiding solution). Starting from the initial solution, the best move still in Δ is applied, until the guiding solution is attained. The best solution found along this trajectory is also considered as a candidate for insertion in the pool and the incumbent is updated. Several alternatives have been considered and combined in recent implementations to explore trajectories connecting X and Y:

- do not apply path relinking at every GRASP iteration, but instead only periodically;
- explore two different trajectories, using first X, then Y as the initial solution;
- explore only one trajectory, starting from either X or Y; and
- do not follow the full trajectory, but instead only part of it.

All these alternatives involve trade-offs between computation time and solution quality. Ribeiro et al. (2002) observed that exploring two different trajectories for each pair $X - Y$ takes approximately twice the time needed to explore only one of them, with very marginal improvements in solution quality. They also observed that if only one trajectory is to be investigated, better solutions are found when path relinking starts from the best among X and Y. Since the neighborhood of the initial solution is much more carefully explored than that of the guiding one, starting from the best of them gives to the algorithm a better chance to investigate with more details the neighborhood of the most promising solution. For the same reason, the best solutions are usually found closer to the initial solution than to the guiding one, allowing pruning the relinking trajectory before the latter is reached. The same findings were also observed on a recent implementation of a GRASP heuristic for a multicommodity flow problem arising from PVC rerouting in frame relay services. Detailed computational results and implementation strategies are described by Resende and Ribeiro (2002).

Path relinking is a quite effective strategy to introduce memory in GRASP, leading to very robust implementations. This is illustrated by the results obtained with the hybrid GRASP with path relinking for the Steiner problem in graphs described in Ribeiro et al. (2002) which in particular improved the best known solutions for 33 out

of the 41 still open problems in series i640 of the SteinLib repository (Voss et al., 2001) on April 6, 2001.

Even though parallelism is not yet systematically used to speed up or to improve the effectiveness of metaheuristics, parallel implementations are very robust and abound in the literature (see e.g. Cung et al. (2001) for a recent survey). Most parallel implementations of GRASP follow the *independent-thread multiple-walk* strategy, based on the distribution of the iterations over the processors.

The efficiency of multiple-walk independent-thread parallel implementations of metaheuristics, running multiple copies of the same sequential algorithm, has been addressed by some authors. A given target value τ for the objective function is broadcasted to all processors which independently run the sequential algorithm. All processors halt immediately after one of them finds a solution with value at least as good as τ. The speedup is given by the ratio between the times needed to find a solution with value at least as good as τ, using respectively the sequential algorithm and the parallel implementation with ρ processors. These speedups are linear for a number of metaheuristics, including simulated annealing, iterated local search, and tabu search. This observation can be explained if the random variable *time to find a solution within some target value* is exponentially distributed (Verhoeven and Aarts, 1995). In this case, the probability of finding a solution within a given target value in time pt with a sequential algorithm is equal to that of finding a solution at least as good as the former in time t using p independent parallel processors, leading to linear speedups. An analogous proposition can be stated for a two parameter (shifted) exponential distribution.

Aiex et al. (2002) have shown experimentally that the solution times for GRASP also have this property, showing that they fit a two-parameter exponential distribution. This result was based on computational experiments involving GRASP procedures applied to 2400 instances of five different problems: maximum independent set, quadratic assignment, graph planarization, maximum weighted satisfiability, and maximum covering. The same result still holds when GRASP is implemented in conjunction with a post-optimization path relinking procedure.

In the case of *multiple-walk cooperative-thread* parallel strategies, the threads running in parallel exchange and share information collected along the trajectories they investigate. One expects not only to speed up the convergence to the best solution but, also, to find better solutions than independent-thread strategies. Cooperative-thread strategies may be implemented using path-relinking, combining elite solutions stored in a central pool with the local optima found by each processor at the end of each GRASP iteration. Canuto et al. (2000, 2001) used path relinking to implement a parallel GRASP for the prize-collecting Steiner tree problem. A similar approach was recently adopted by Aiex et al. (2000) for the 3-index assignment problem. Each processor, upon completing its iterations, applies path relinking to pairs of elite solutions stored in a pool, and each processor keeps its own local pool of elite solutions. The strategy used in Canuto (2000) is truly cooperative, since pairs of elite solutions from a centralized unique central pool are distributed to the processors which perform path relinking in parallel. Computational results obtained with implementations using MPI and running on a cluster of 32 Pentium II-400 processors and on a SGI Challenge computer with 28 196-MHz MIPS R10000 processor (Aiex et al., 2000) show linear speedups and further illustrate the effectiveness of path relinking procedures used in conjunction with GRASP to improve the quality of the solutions found by the latter.

ACKNOWLEDGEMENTS

Research by F. Glover and M. Laguna was partially supported by the U.S. Office of Naval Research grant N00014-02-0151, and research by R. Martí partially supported by the Ministerio de Ciencia y Tecnología of Spain: TIC2000-1750-C06-01.

REFERENCES

Aiex, R.M., M.G.C. Resende, P.M. Pardalos, and G. Toraldo (2000) "GRASP with path relinking for the three-index assignment problem," submitted for publication.

Aiex, R.M., M.G.C. Resende, and C.C. Ribeiro (2002) "Probability distribution of solution time in GRASP: An experimental investigation", *Journal of Heuristics* 8, 343-373.

Alvarez A., González J.L., De Alba K. (2001a) "Un Algoritmo de Búsqueda para un Problema de Red Capacitada Multiproducto," Proceedings of 3rd International Meeting of Computational Sciences. (Ed. : C. Zozaya, M. Mejía, P. Noriega, A. Sánchez) . Aguascalientes México, pp. 105-114.

Alvarez A., González J.L., De Alba K. (2001b) "Scatter Search for the Multicommodity Capacitated Network Design Problem," Proceedings of the 6th Annual International Conference on Industrial Engineering – Theory, Applications and Practice. San Francisco, CA, USA.

Beausoleil, R.P. (2001) "Multiple Criteria Scatter Search," 4th Metaheuritics International Congress, Porto, Portugal, pp. 539-543.

Bulut, G. (2001) "Robust Multi-Scenario Optimization Of An Air Expeditionary Force Force Structure Applying Scatter Search To The Combat Forces Assessment Model," Masters Thesis, Department of Operational Sciences, Air Force Institute of Technology, AFIT/GOR/ENS/01M-05.

Campos, V., Laguna, M., and Martí, R., (1999) "Scatter Search for the Linear Ordering Problem," in *New Ideas in Optimization*, Eds. David Corne, Marco Dorigo and Fred Glover, McGraw-Hill, pp. 331-340.

Canuto, S.A. (2000) *Local search for the prize-collecting Steiner tree problem* (in Portuguese), M.Sc. Dissertation, Department of Computer Science, Catholic University of Rio de Janeiro.

Canuto, S.A., M.G.C. Resende, and C.C. Ribeiro, (2001) "Local search with perturbations for the prize-collecting Steiner tree problem in graphs," *Networks* 38, pp. 50-58.

Cavique L., Rego C., Themido I. (2001) "A Scatter Search Algorithm for the Maximum Clique Problem," *Essays and Surveys in Metaheuristics*, Kluwer Academic Publishers.

Chanas, S. and P. Kobylanski (1996) "A New Heuristic Algorithm Solving the Linear Ordering Problem," Computational Optimization and Applications, vol. 6, pp. 191-205.

Crowston, W.B., G.L. Thompson and J.D. Trawick (1963). "Probabilistic Learning Combinations of Local Job Shop Scheduling Rules," Chapters II and III, ONR Research Memorandum No. 117, GSIA, Carnegie Mellon University, Pittsburgh, PA.

Cung, V.-D., T. Mautor, P. Michelon and A. Tavares (1997) "A scatter search based approach for the quadratic assignment problem," in T. Bäck, Z. Michalewicz and X. Yao, (Eds.), Proceedings of IEEE-ICEC-EPS'97, IEEE International Conference on Evolutionary Computation and Evolutionary Programming Conference, pp. 165–170, Indianapolis.

Cung, V.D., S.L. Martins, C.C. Ribeiro, and C. Roucairol (2001) “Strategies for the parallel implementation of metaheuristics”, in *Essays and Surveys in Metaheuristics* (C.C. Ribeiro and P. Hansen, eds.), pages 263-308, Kluwer.

Delgado, C., M. Laguna and J. Pacheco (2002) " Minimizing Labor Requirements in a Periodic Vehicle Loading Problem," University of Burgos, Spain.

Dorne, R. and Hao, J.-K. (1998) “Tabu Search for Graph Coloring, T-Colorings and Set T-Colorings,” in *Meta-heuristics: Advances and Trends in Local Search Paradigms for Optimization*, S. Voss, S. Martello, I.H. Osman and C. Roucairol (Eds.), Kluwer, pp. 77-92.

Eades, P. and D. Kelly (1986) “Heuristics for Drawing 2-Layered Networks,” *Ars Combinatoria,* Vol. 21, pp. 89-98.

Festa P. and Resendee M.G.C (2001) “GRASP: an annotated bibliography,” In: P.Hansen and C. Riberio co-editors, Essays and Surveys on Meta-Heuristics, Kluwer Academic Publishers, Boston USA.

Fleurent C. and F. Glover (1999) Improved constructive multi-start strategies for the quadratic assignment problem using adaptive memory. INFORMS J. On computing, 11, pp. 198-204.

Funabiki, N. and Higashino, T. (2000) “A Minimal-State Processing Search Algorithm for Graph Colorings Problems”, *IEICE Transactions on Fundamentals*, E83-A(7), pp. 1420-1430.

Galinier, P. and Hao, J.-K. (1999) “Hybrid Evolutionary Algorithms for Graph Coloring”, *Journal of Combinatorial Optimization*, 3(4), pp. 379-397.

García, F. , Melián B., Moreno J.A. and Moreno J.M. (2001), “Hybrid MetaHeuristics based on the Scatter Search,” Proceeding of EUNITE 2001. pp. 479-485. ISBN: 3-89653-916-7 (European Symposium on Intelligent Technologies, Hybriis Systems and their implementation on Smart Adaptive Systems. December 13-14, 2001. Puerto de la Cruz, Tenerife, SPAIN.]

Garey M., Johnson D. (1979). *Computers and Intractability. A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company. New York.

Ghamlouche, I., Crainic, T.G., Gendreau, M. (2002), "Path Relinking, Cycle-Based Neighbourhoods and Capacitated Multicommodity Network Design," Publication CRT-2002-01, Centre de recherche sur les transports, Université de Montréal.

Ghamlouche, I., Crainic, T.G., Gendreau, M. (2001), "Cycle-based Neighbourhoods for Fixed-Charge Capacitated Multicommodity Network Design", Publication CRT-2001-01, Centre de recherche sur les transports, Université de Montréal.

Glover, F. (1963) “Parametric Combinations of Local Job Shop Scheduling Rules,” Chapter IV, ONR Research Memorandum No. 117, GSIA, Carnegie Mellon University, Pittsburgh, PA.

Glover, F. (1965) “A Multiphase Dual Algorithm for the Zero-One Integer Programming Problem,” *Operations Research*, Vol 13, No 6, pp. 879-919.

Glover, F. (1968) “Surrogate Constraints,” *Operations Research*, Vol. 16. No. 4, pp. 741-749.

Glover, F. (1977) “Heuristics for Integer Programming Using Surrogate Constraints,” *Decision Sciences*, Vol 8, No 1, pp 156-166.

Glover, F. (1994) "Tabu Search for Nonlinear and Parametric Optimization (with Links to Genetic Algorithms)," *Discrete Applied Mathematics* 49, pp. 231-255.

Glover, F. (1995) "Scatter Search and Star Paths: Beyond the Genetic Metaphor," *OR Spektrum*, vol. 17, pp. 125-137.

Glover F. (1997) Tabu Search and Adaptive memory programming- advances, applications and challenges. In. R. Barr, Helgason, and Kennington, Co-editors, Advances in Meta-heuristics, Optimization and Stochastic Modeling Techniques, Pages 1-175, Kluwer Academic Publishers, Boston, USA.

Glover, F. (1998) "A Template for Scatter Search and Path Relinking," in *Artificial Evolution, Lecture Notes in Computer Science* 1363, J.-K. Hao, E. Lutton, E. Ronald , M. Schoenauer and D. Snyers (Eds.), Springer, pp. 3-51.

Glover, F. and M. Laguna (1997) *Tabu Search,* Kluwer Academic Publishers, Boston.

Glover, F., A. Løkketangen and D. L. Woodruff. (2000) "Scatter Search to Generate Diverse MIP Solutions". In: *OR Computing Tools for Modeling, Optimization and Simulation: Interfaces in Computer Science and Operations Research*, Eds. M. Laguna and J.L. González-Velarde, pp. 299 – 317.

Grabowski, J. and M. Wodecki (2001) "A new very fast tabu search algorithm for the job shop problem," Preprint 21/2001, Instytut Cybernetyki Technicznej Politechniki Wroclawskiej, Wroclaw.

Greistorfer, P. (1995) "Computational experiments with heuristics for a capacitated arc routing problem," in U. Derigs, A. Bachem and A. Drexl, (Eds.), Operations Research Proceedings 1994, pp. 185–190. Springer-Verlag, Berlin.

Greistorfer, P. (2001a) "A tabu scatter search metaheuristic for the arc routing problem", Computers & Industrial Engineering (forthcoming).

Greistorfer, P. (2001b) "Testing population designs". 4th Metaheuristics International Conference (MIC'2001), pp. 713–717, Porto. (17 pages submitted as "Experimental pool design").

Hamiez, J.P., and Hao, J.K. (2001) "Scatter Search for Graph Coloring," to appear in the *LNCS* series (Springer).

Herrmann J.W., Ioannou G., Minis I., Proth J.M. (1996) A Dual Ascent Approach to the Fixed-Charge Capacitated Network Design Problem. European Journal of Operational Research 95, 476-490.

Hill, R. R. and G. McIntyre. (2000) A Methodology for Robust, Multi-Scenario Optimization. *Phalanx*, Vol 33, No. 3.

Johnson, D.S. and Trick., M.A. (Eds., 1996) *Cliques, Coloring, and Satisfiability: 2nd DIMACS Implementation Challenge, 1993*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 26, American Mathematical Society.

Knuth, D. E. (1993) *The Stanford GraphBase: A Platform for Combinatorial Computing*, Addison Wesley, New York.

Laguna, M. (1999) "Scatter Search", to appear in *Handbook of Applied Optimization*, P. M. Pardalos and M. G. C. Resende (Eds.), Oxford Academic Press.

Laguna, M. and Martí, R. (1999) " GRASP and Path Relinking for 2-layer straight line crossing minimization," *INFORMS Journal on Computing*, 11, 1, pp. 44-52.

Laguna and Martí (2002) "The OptQuest Callable Library," *Optimization Software Class Libraries*, Eds. S.Voss and D.L. Woodruff, Kluwer, Boston.

Laguna, M., R. Martí and V. Campos (1999) "Intensification and Diversification with Elite Tabu Search Solutions for the Linear Ordering Problem," *Computers and Operations Research*, 26, PP. 1217-1230

Løkketangen, A. and Woodruff. (2000) "Integrating Pivot Based Search with Branch and Bound for Binary MIP's". In: *Control and Cybernetics, Special Issue on Tabu Search*. Vol. 29, No.3, pp 741 – 760.

Løkketangen, A, and F. Glover (1998) "Solving zero-one mixed integer programming problems using tabu search". *European Journal of Operational Research,* Vol 106, pp 624-658.

Løkketangen, A. and F. Glover (1995) "Tabu Search for Zero/One Mixed Integer Programming with Advanced Level Strategies and Learning". *International Journal of Operations and and Quantitative Management*, Vol. 1 No. 2. pp 89 - 109.

Løkketangen, A. and F. Glover (1996) "Probabilistic Move Selection inTabu Search for 0/1 Mixed Integer Programming Problems". In: *Metaheuristics: Theory and Applications*, by Kluwer, mar -96. An earlier version is in the conference proceedings from MIC '95.

Løkketangen, A. and F. Glover (1999) "Candidate List and Exploration Strategies for Solving 0/1 MIP Problems using a Pivot Neighborhood". In: *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, Eds. S. Voß, S. Martello, I.H. Osman and C. Roucairol, Kluwer Academic Publishers, pp 141 – 155.

Magnanti T., and R. Wong (1984) Network Design and Transportation Planning: Models and Algorithms. Transportation Science 18, 1–55.

Martí R. (1998) "A Tabu Search Algorithm for The Bipartite Drawing Problem", *European Journal of Operational Research*, 106, 558-569.

Martí R. and M. Laguna (1997) "Heuristics and Metaheuristics for 2-Layer Straight Line Crossing Minimization," to appear in *Discrete and Applied Mathematics*.

Martí, R. L. Lourenço and M. Laguna (2000) "*Assigning Proctors to Exams with Scatter Search*", in Computing Tools for Modeling, Optimization and Simulation, M. Laguna and J.L. González Velarde (eds.), Kluwer Academic Publishers, pp. 215-228.

Martí, R. L. Lourenço and M. Laguna (2001) "*Assigning Proctors to Exams with Scatter Search (second part)*", Economic Working Papers Series, Department of Economics and Business, Universitat Pompeu Fabra, n. 534.

Morgenstern, C.A. (1996) "Distributed Coloration Neighborhood Search", in (Johnson and Trick, 1996), pp. 335-357.

Nowicki E., and C. Smutnicki (1996) "A fast tabu search algorithm for the job-shop problem," Management Science 42(6), pp. 797-813.

Nowicki E., and C. Smutnicki (2001a). "New ideas in TS for job-shop scheduling," Technical Report 50/2001. (submitted to book "Adaptive Memory and Evolution: Tabu Search and Scatter Search", eds. C. Rego and B. Alidaee, Kluwer Academic Publisher).

Nowicki E., and C. Smutnicki (2001b) “New tools to solve the job-shop problem,” Technical Report 51.

Osman, I.H. Meta-heuristics: A general framework. In Proceedings of the Workshop on "algorithm engineering as a new paradigm: a challenge to hard computation problems" October 30-Nov 2nd 2000, pages 117-118, Research Institute for Mathematical Science, Kyoto University, Japan.

Osman, I.H. and Samad Ahmadi, Guided Construction search Meta-Heuristics for the Capacitated Clustering Problem. Working Paper, School of Business, American University of Beirut, August 2001.

Osman, I.H., Capacitated clustering problems by hybrid simulated annealing and tabu search, International Transactions in Operational Research I (1994) 317-336.

Rego, Cesar and Pedro Leão (2000) "A Scatter Search Tutorial for Graph-Based Permutation Problems," Hearin Center for Enterprise Science, U of Mississippi, Report Number: HCES-10-00.

Reinelt, G. (1985) *The Linear Ordering Problem: Algorithms and Applications*, Research and Exposition in Mathematics, Vol. 8, H. H. Hofmann and R. Wille (Eds.), Heldermann Verlag Berlin.

Resende, M.G.C. and C.C. Ribeiro (2001) “A GRASP with path relinking for permanent virtual circuit routing”, Research Report, submitted for publication.

Resende, M.G.C. and C.C. Ribeiro (2002) “GRASP”, to appear in *State-of-the-Art Handbook of Metaheuristics* (F. Glover and G. Kochenberger, eds.), Kluwer.

Ribeiro, C.C., E.~Uchoa, and R.F. Werneck (2002) “A hybrid GRASP with perturbations for the Steiner problem in graphs”, to appear in *INFORMS Journal on Computing*.

Rusell, R.A. and Igo, W. (1979) “An assignment routing problem”. *Networks* 9, 1-17.

Souza, M.C., C. Duhamel, and C.C. Ribeiro (2002) “A GRASP heuristic using a path-based local search for the capacitated minimum spanning tree problem”, Research Report, submitted for publication.

Valls, V., S. Quintanilla and F. Ballestín (2001), “A Population Based Approach to the Resource Constrained Project Scheduling”, TR06-2001, Departamento de Estadística e Investigación Operativa, Facultad de Matemáticas, Universitat de València (Spain).

Verhoeven, M.G.A. and E.H.L. Aarts (1995) “Parallel local search”, *Journal of Heuristics* 1, 43-65.

Voss, S., A. Martin, and T. Koch (2001) “SteinLib testdata library”, online document at http://elib.zib.de/steinlib/steinlib.html.

Woodruff, D.L. (1998) “Proposals for Chunking and Tabu Search,” *European J. Oper. Res.,* vol. 106, pp. 585-598.

Woodruff, D.L. (2001) “General Purpose Metrics for Solution Variety,” Technical Report, Graduate School of Management, UC Davis, Davis CA 95616.

Woodruff, D.L. (1996) “Chunking Applied to Reactive Tabu Search”, in Metaheuristics: Theory and Applications, I.H. Osman and J.P. Kelly, eds., pp 555 – 570.