

Chapter 13

PARALLEL COMPUTATION, CO-OPERATION, TABU SEARCH

Teodor Gabriel Crainic

Département de management et technologie, École des Sciences de la Gestion, Université du Québec à Montréal and Centre de recherche sur les transports, Université de Montréal, theo@crt.umontreal.ca

Abstract We present strategies to design parallel tabu search algorithms and survey developments and results in the area. In the second part of the paper, we focus on multi-search strategies. We discuss general design and implementation principles, point out a number of challenges and pitfalls, and identify trends and promising research directions.

Keywords: Parallel Computation, Parallelization Strategies, Co-operative Search, Tabu Search, Scatter Search, Path Relinking

1. Introduction

Parallel computation is increasingly acknowledged as a natural complement to problem solving strategies, particularly when applied to large-scale, hard formulations and instances. The field of meta-heuristics is no exception to this trend and the last ten to fifteen years have witnessed a continuously stronger stream of important developments, most of which targeted tabu search, genetic algorithms, and simulated annealing methods (e.g., the surveys of Crainic and Toulouse 1998, 2003, Cung *et al.* 2002, Holmqvist, Migdalas, and Pardalos 1997, Pardalos *et al.* 1995, and Verhoeven and Aarts 1995). While a number of these efforts addressed particular problem instances and yielded tailor-made procedures, most proposed strategies of a more general nature, at least within the framework of the underlying sequential search methodology. These strategies are the object of this paper.

Parallel computing methods aim to solve a given problem quicker than the corresponding sequential method. In meta-heuristic terms, this goal may be stated as either “accelerate the search for a comparable solution quality” or

“broaden the search”, that is, obtain better solutions for a comparable computation effort or, at least, for the same wall-clock time. Parallel meta-heuristic search methods should also be more robust than sequential methods with respect to the characteristics of the problem instances they encounter. Parallel implementations are thus expected not only to identify better solutions but also to do it consistently over diverse sets of problem instances without excessive calibration efforts.

The review of the literature reveals that, beyond the idiosyncrasies of each implementation that follow from the particular problem instance addressed, there are only a limited number of general strategies that may be used to build parallel meta-heuristics. A number of these strategies are well-understood by now, others are still a fertile ground for research. Together, they enhance our ability to address hard and large-scale problems. In this paper we focus on tabu search. The objectives are to briefly review parallel strategies and developments for tabu search, to discuss in somewhat more details co-operative methods that are the subject of most contemporary efforts, and to identify promising research directions.

The paper is organized as follows. Section 2 describes parallel strategies and reviews parallel tabu search methods proposed in the literature. Section 3 describes the main multi-search strategies and discusses the challenges associated to the development of successful co-operative methods. It also presents a number of ideas on how parallel strategies could be applied to some advanced tabu search methods, particularly path relinking and scatter search. Section 4 sums up the paper with a number of interesting challenges and research directions for parallel tabu search.

2. Parallel Tabu Search

Meta-heuristics have been defined as master strategies (heuristics) to guide and modify other heuristics to avoid getting trapped in local optima or sequences of visited solutions (*cycling*), produce solutions beyond those normally identified by local search heuristics, and provide reasonable assurance that the search has not overlooked promising regions (Glover 1986, Glover and Laguna 1993).

Tabu search, similarly to most meta-heuristics, is an *improving* iterative procedure that *moves* from a given solution to a solution in its *neighbourhood* that is better in terms of the objective function value (or some other measure based on the solution characteristics). Thus, at each iteration, a *local search* procedure (the “original” heuristic) identifies and evaluates solutions in the neighbourhood of the current solution, selects the best one relative to given criteria, and implements the transformations required to establish the selected solution as the current one. Inferior quality solutions encountered during neighbourhood explorations may be accepted as a strategy to move away from local optima. In

theory, the procedure iterates until no further improvement is possible. In actual implementations, a more restrictive stopping criteria is used: total number of iterations, relative improvement over a number of iterations, etc.

Memory and *memory hierarchy* are major concepts in tabu search, as memory-based strategies are used to guide the procedure into various search phases, possibly exploring different neighbourhoods. Short-term *tabu status* memories record recent visited solutions or their attributes to avoid cycling due to the repetition or inversion of recent actions. The tabu status of a move may be lifted if testing it against an *aspiration criterion* signals the discovery of a high quality solution (typically, the best one encountered so far). Medium to long-term memory structures record various informations and statistics relative to the solutions already encountered (e.g., frequency of certain attributes in the best solutions) to “learn” about the solution space and guide the search. *Intensification* of the search around good solutions and its *diversification* towards regions of the solution space not yet explored are two main ingredients of tabu search. These two types of moves are based on medium and long-term memories, are implemented using specific neighbourhoods, and may involve quite complex solution transformations. More details on the basic and advanced features of tabu search may be found in Glover (1986, 1989, 1990, 1996) and Glover and Laguna (1993, 1997).

Tabu search has proved a fertile ground for innovation and experimentation in the area of parallel meta-heuristics. Most parallel strategies encountered in the literature have been applied to tabu search for a variety of applications and a number of interesting parallelization concepts have been introduced while developing parallel tabu search methods (Crainic and Toulouse 2003).

Crainic, Toulouse, and Gendreau (1997) introduced in 1993 what is still the most comprehensive taxonomy of parallel tabu search methods. The classification has three dimensions (Figure 13.1). The first dimension, *Search Control Cardinality*, explicitly examines how the global search is controlled: either by a single process (as in master-slave implementations) or collegially by several processes that may collaborate or not. The two alternatives are identified as *l-control (lC)* and *p-control (pC)*, respectively. The classes of the second dimension indicate the *Search Differentiation*: do search threads start from the same or different solutions and do they make use of the same or different search strategies? The four cases considered are: *SPSS, Same initial Point, Same search Strategy*; *SPDS, Same initial Point, Different search Strategies*; *MPSS, Multiple initial Points, Same search Strategies*; *MPDS, Multiple initial Points, Different search Strategies* (see also Voß 1993). Finally, the dimension relative to the type of *Search Control and Communications* addresses the issue of how information is exchanged.

In parallel computing, one generally refers to *synchronous* and *asynchronous* communications. In the former case, all processes have to stop and engage in

some form of communication and information exchange at moments (number of iterations, time intervals, specified algorithmic stages, etc.) exogenously determined, either hard-coded or determined by a control (master) process. In the latter case, each process is in charge of its own search, as well as of establishing communications with the other processes, and the global search terminates once each individual search stops. To reflect more adequately the quantity and quality of the information exchanged and shared, as well as the additional knowledge derived from these exchanges (if any), we refine these notions and define four classes of Search Control and Communication strategies, *Rigid (RS)* and *Knowledge Synchronization (KS)* and, symmetrically, *Collegial (C)* and *Knowledge Collegial (KC)*. Crainic, Toulouse, and Gendreau (1997) detail the taxonomy and use it to analyze the parallel methods proposed in the literature up to that time.

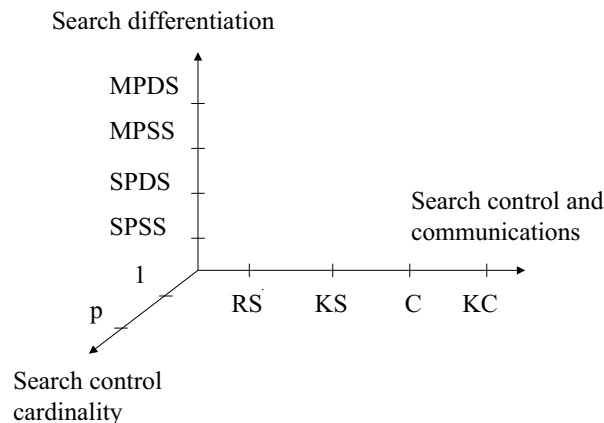


Figure 13.1. Method-oriented Taxonomy

Typically, 1-control strategies implement a classical master-slave approach that aims solely to accelerate the search. Here, a “master” processor executes a synchronous tabu search procedure but dispatches computing-intensive tasks to be executed in parallel by “slave” processes. The master receives and processes the information resulting from the slave operations, selects and implements moves, gathers all the information generated during the tabu exploration, updates the memories, and decides whether to activate different search strategies or stop the search.

In the context of tabu search, the operation most widely targeted in such approaches is the neighbourhood evaluation. At each iteration, the possible moves in the neighbourhood of the current solution are partitioned into as many sets as the number of available processors and the evaluation is carried out in parallel by slave processes. This 1C/RS/SPSS strategy yielded very interesting results for problems with large neighbourhoods and relatively small computing efforts required to evaluate and perform a given move, such as the *quadratic assignment* (QAP: Chakrapani and Skorin-Kapov 1992, 1993b, 1995, Taillard 1991, 1993b), *travelling salesman* (TSP: Chakrapani and Skorin-Kapov 1993a) and *vehicle routing* (VRP: Garcia, Potvin, and Rousseau 1994) problems. For the same quality of solution, near-linear speedups are reported using a relatively small number of processors. Moreover, historically, this approach permitted improvements to the best-known solutions to several problem instances proposed in the literature.

Implementations of the *sequential fan candidate list* strategy (Glover, Taillard, and de Werra 1993, Glover and Laguna 1997), also known as *look ahead* or *probing* approaches, allow slave processes to perform a number of iterations before synchronization and the selection of the best neighbouring solution from which the next iteration is initiated. Probing strategies thus belong to the 1-control, knowledge synchronous (1C/KS) class and may use any of the four search differentiation models identified earlier on. The only parallel implementation of this strategy the author is aware of is to be found in the comparative study of several synchronous tabu search parallelizations performed by Crainic, Toulouse, and Gendreau (1995a) for the location-allocation problem with balancing requirements. In this study, the authors implemented and compared a 1C/RS/SPSS and a 1C/KS/SPSS method. The second performed marginally better. However, both methods were outperformed by p-control implementations that attempt a more thorough exploration of the solution space. These results also emphasize that performance is less interesting for the low-level parallelism offered by single control, synchronous parallel strategies when the time required by one serial iteration is relatively important compared to the total solution time and only a few hundred moves may be executed in a reasonable computing time (compared to the tens of thousands performed by a typical VRP tabu search procedure).

Domain decomposition is another major parallelization strategy. The fundamental idea is simple: Partition the feasible domain of the problem in several (disjoint) sets and execute the contemplated meta-heuristic on each subset, thus accelerating the global search. Two cases may then be encountered. Either the solution of each search thread is a complete solution to the problem in hand or a partial solution only, in which case a complete solution has to be reconstructed. In both cases, however, each search process has had access to a part of the domain only (especially when the partition is strictly enforced during

the search). Then, to somewhat increase the thoroughness of the search, the partition is modified and the search is started anew. This process is repeated a number of times.

The “natural” implementation of domain decomposition is through a 1C/KS master-slave scheme with any of the search differentiation strategy. (Notice that different meta-heuristics may be applied to different sets and that, by definition, each search starts from a different initial point.) The master determines the partition, synchronizes slave processes, reconstructs solutions (when required), and determines stopping conditions. Slave processes perform the search on their assigned partitions. Such approaches have proved successful for problems for which numerous iterations may be performed in a relatively short time and restarting the method with several different partitions does not require unreasonable computational efforts (e.g., Fiechter 1994 for the *TSP*, Porto and Ribeiro 1995, 1996, and Porto, Kitajima, and Ribeiro 2000 for the task scheduling problem on heterogeneous systems).

The same idea may also be implemented in a pC/KS framework, however. In this case, processors stop at pre-determined moments to exchange partial solutions and build whole ones, modify the partition, verify the stopping conditions. From a computer programming point of view, one of the processors may assume co-ordination tasks, but this does not change the particular nature of the algorithmic design. Taillard’s (1993a) early tabu search for VRP follows this pC/KS/MPSS paradigm. The domain is partitioned and vehicles are allocated to the resulting regions. Once the initial partition is performed, each subproblem is solved by an independent tabu search. All processors stop after a number of iterations that varies according to the total number of iterations already performed. The partition is then modified by an information exchange phase, during which tours, undelivered cities, and empty vehicles are exchanged between adjacent processors (corresponding to neighbouring regions).

Partition approaches did allow to address successfully a number of problem instances. The synchronization inherent in the design of the strategy hinder its performance, however. Indeed, in most cases, results obtained by using partition strategies have been improved by methods that launch several search threads to simultaneously explore the solution space with various degrees of co-operation. In fact, enjoying the benefit of hindsight, the main contribution of the Taillard (1993a) paper is to mark the evolution towards one of the most successful sequential meta-heuristics for the VRP, the *adaptive memory* tabu search (Rochat and Taillard 1995, Glover 1996).

According to an adaptive memory approach applied to the VRP, cities are initially separated into several subsets, and routes are built using a construction heuristic. Initial routes are then stored in a structure called an *adaptive memory*. Then, a combination procedure builds a complete solution using the routes in the memory and the solution is further improved using a tabu search method.

The routes of “good” solutions are then deposited into the same memory, which thus adapts to reflect the current state of knowledge of the search. The process then re-starts with a new solution built from the routes stored in the adaptive memory. The method stops when a pre-specified number of calls to the adaptive memory have been performed. This approach clearly implements the principles of decomposition using a serial procedure and has now been successfully applied to other problem classes. However, interestingly, most parallel applications of this approach are now found in co-operative multi-thread strategies.

Multi-thread or *multi-search* parallelizations for tabu search follow the same basic pattern: p threads search through the same solution space, starting from possibly different initial solutions and using possibly different tabu (or other) search strategies. Historically, independent and synchronous co-operative multi-thread methods were proposed first. Currently, asynchronous procedures are being generally developed. One also observes an increased interest in issues related to the definition and modelling of co-operation (Section 3).

Independent multi-searches belong to the pC/RS class of the taxonomy. Most implementations start several independent search processes from different, randomly generated, initial configurations. No attempt is made to take advantage of the multiple threads running in parallel other than to identify the best overall solution once all processes stop. This definitively earns independent search strategies their Rigid Synchronization classification. (Note that, in general, the implementations designate a processor to collect the information and verify stopping criteria.) Battiti and Tecchiolli (1992, for the QAP), Taillard (the main study is found in his 1994 paper on parallel tabu methods for job shop scheduling problems), and others studied and empirically established the efficiency of such procedures when compared to the best heuristics proposed at the time for their respective problems. This parallelization of the classic sequential multi-start heuristic is easy to implement and may offer satisfactory results. Co-operative strategies often offer superior performance, however (e.g., Crainic, Toulouse, and Gendreau 1995b, Crainic and Gendreau 2002).

pC/KS strategies are also generally implemented in a master-slave setting but attempt to take advantage of the parallel exploration by synchronizing processors at pre-determined intervals. The master process then collects information and usually restarts the search from the best solution (Malek *et al.* 1989 for the TSP, Rego and Roucairol 1996 for the VRP using ejection chains). De Falco *et al.* (1994 for the QAP), and De Falco, Del Balio, and Tarantino (1995 for the mapping problem) attempted to overcome the limitations of the master-slave setting. When each search thread terminates its local search, they synchronize and best solutions are exchanged between processes that run on neighbouring processors. Good performance and results were reported by all authors.

Asynchronous co-operative multi-thread search methods belong to the pC/C or pC/KC classes of the taxonomy according to the quantity and quality of

the information exchanged and, eventually, on the “new” knowledge inferred based on these exchanges. Most such developments use some form of *memory* for inter-thread communications (the term *blackboard* is also used sometimes). Each individual search thread starts from (usually) a different initial solution and generally follows a different search strategy. Exchanges are performed asynchronously and through the memory. Memories recording the performance of individual solutions, solution components, or even search threads may be added to the pool and statics may be gradually built. Moreover, various procedures may also be added to the pool to attempt to extract information or to create new informations and solutions based on the solutions exchanged. Co-operative multi-thread strategies implementing such methods belong to the pC/KC class.

One may classify co-operative multi-thread search methods according to the type of information stored in the central memory: complete or partial solutions. In the latter case, one often refers to *adaptive memory* strategies, while *central memory*, *pool of solutions*, *solution warehouse* or, even, *reference set* are terms used for the former. These methods have proved extremely successful on a broad range of problem types: real-time routing and vehicle dispatching (Gendreau *et al.* 1999), VRP with time windows (Taillard *et al.* 1997, Badeau *et al.* 1997, Schulze and Fahle 1999, Le Bouthiller and Crainic 2004), multicommodity location with balancing requirements (Crainic, Toulouse, and Gendreau 1995b), fixed cost, capacitated, multicommodity network design (Crainic and Gendreau 2002), and partitioning of integrated circuits for logical testing (Andreatta and Ribeiro 1994, Aiex *et al.* 1996, 1998, and Martins, Ribeiro, and Rodriguez 1996). One must notice, however, that the selection and the utilization of the information exchanged significantly impacts the performance of co-operating procedures. Co-operation mechanisms have to be carefully designed (Toulouse, Crainic, and Gendreau 1996, Toulouse, Thulasiraman, and Glover 1999). We return to these issues in the next section.

Although introduced for tabu search, the classification used in this paper applies to many other classes of meta-heuristics and refines the criteria based on the impact of parallelization on the search trajectory (e.g., Crainic and Toulouse 2003). Thus, it may form the basis for a comprehensive taxonomy of parallel meta-heuristics. This task, however, is beyond the scope of this paper.

3. Co-operation, Path Relinking, Scatter Search

A trend emerges from the collective efforts dedicated to parallel meta-heuristics in general and tabu search in particular. The main methodological focus moved from the low-level parallelism (e.g., the 1C/RS methods) of the initial developments, through a phase of domain decomposition approaches, on to independent search methods and co-operation (and hybridization). Of course, each type of parallelization strategy may play an important role given the particular problem

class and instance. In this section, we focus on co-operation, however, since it appears to offer both the most promising avenues for superior performances and the biggest challenges in terms of algorithm design.

Co-operative multi-thread tabu search methods launch several independent searches and implement information-exchange mechanisms among these threads. The general objective is to exchange *meaningful* information in a *timely* manner such that the global parallel search achieves a better performance than the simple concatenation of the results of the individual methods. Performance is measured in terms of computing time and solution quality (Barr and Hickman 1993, Crainic and Toulouse 2003).

As mentioned already, independent search methods implement multi-start sequential heuristics and, therefore, cannot achieve both performance objectives. Computation time is indeed reduced, but the same solution is reached. To overcome this limitation, Knowledge Synchronous pC strategies have been implemented where information is exchanged once all threads have completed their searches. The best overall solution is the only information exchanged and it serves as departure point for a new round of independent searches. The method is simple to implement and authors report good results. Computing times soar, however, and it would be interesting to thoroughly compare solution quality performance with co-operative methods. The limited evidence to this effect comes from applications to vehicle routing formulations and seem to indicate that co-operative strategies perform better in general.

Two other issues should be investigated in relation to these pC/KS strategies. First, the implementations reported in the literature start *new* searches following synchronization, cleaning up all memories. Given the importance of memories for tabu search, one wonders whether precious information is lost in the process. An investigation into what information to pass from one independent search phase to the next, how to use it, and its impact on performance might prove very interesting. The second issue has to do with the fact that, up to now, the only information exchanged when processes synchronize is the best solution. Yet, it has been shown that for co-operative methods this strategy may be counter-productive by concentrating most (all) threads on the same region of the solution space. This is an issue for further study as well.

Co-operation strategies require careful design as well as resources to manage the exchanges and process the data. A number of fundamental issues have to be addressed when designing co-operative parallel strategies for meta-heuristics (Toulouse, Crainic, and Gendreau 1996):

- What information is exchanged?
- Between what processes it is exchanged?
- When is information exchanged?

- How is it exchanged?

- How is the imported data used?

It is beyond the scope of this paper to address in detail all these issues. The reader may refer to the papers mentioned in this section for a more thorough description. It may be of interest, however, to sum up some of the pitfalls and less-than-winning strategies that have been identified in the literature so far.

Synchronous co-operative implementations tend to show a poorer performance compared to asynchronous and independent searches, especially when synchronization points are pre-determined (as in most current implementations). Such strategies display large computation overheads and a loss of efficiency. This is due to the obligation to wait for all processes to reach the synchronization point. The pre-definition of synchronization points also makes these strategies much less reactive to the particular problem instance and the progress of the search than asynchronous approaches and methods.

Asynchronous strategies are increasingly being proposed. Not all approaches offer the potential for success, however. To illustrate some of the issues, consider the following generic “broadcast and replace” strategy. When a search thread improves its local best solution, a message is sent to all other processes to stop their own exploration and import the broadcast solution. Then, if the imported solution is better than the local one, the search is restarted with the new solution. Several variants of this approach may be found in the literature. It has been observed, however, that interrupting the “local search” phase does not yield good results. The main reason is that one no longer performs a thorough search based on the local observation of the solution space. Rather, one transforms the tabu search into random search. This phenomenon is particularly disruptive if one permits the initial local search phase to be interrupted.

Stated in more general terms, the previous observation says that co-operative meta-heuristics with unrestricted access to shared knowledge may experience serious premature “convergence” difficulties, especially when the shared knowledge reduces to one solution only (the overall best or the new best from a given thread). This is due to a combination of factors: There is no global history or knowledge relative to the trajectory of the parallel search and thus each process has a (very) partial view of the entire search; Threads often use similar criteria to access the (same) “best” solution; Each process may broadcast a new best solution after a few moves only and thus disseminate information where only part of the solution has been properly evaluated. The contents of shared data tend then to become stable and biased by the best moves of the most performing threads and one observes a premature convergence of the dynamic search process. Moreover, the phenomenon may be observed whether one initializes the local memories following the import of an external solution or not (Toulouse

et al. 1998, Toulouse, Crainic, and Sansó 1999, 2004, Toulouse, Crainic, and Thulasiraman 2000).

Toulouse, Thulasiraman, and Glover (1999; see also Toulouse, Glover, and Thulasiraman 1998) proposed a new co-operation mechanism that attempts to address these challenges. The mechanism is called *multi-level co-operative search* and is based on the principle of controlled diffusion of information. Each search process works at a different level of aggregation of the original problem (one processor works on the original problem; the aggregation scheme ensures that a feasible solution at a level is feasible at the more disaggregated levels). Each search communicates exclusively with the processes working on the immediate higher and lower aggregation levels. Improved solutions are exchanged asynchronously at various moments dynamically determined by each process according to its own logic, status, and search history. An incoming solution will not be transmitted further until a number of iterations have been performed. The approach has proven very successful for graph partitioning problems (Ouyang *et al.* 2000, 2000a). Significant research is needed, however, to explore the various possibilities of the mechanisms, including the role of local and global memories.

Adaptive and *central* memory strategies also aim to overcome the limitations of straightforward co-operation by implementing mechanisms to build knowledge regarding the global search. This knowledge is then used to guide the exchange of information and, thus, to impact the trajectory of each individual process. The two approaches have much in common. In both approaches, there is no communication among the individual search processes: communication takes place exclusively between the memory process and individual search threads. Both approaches control when communication takes place (not very often and not during local search phases) and assign this responsibility to the individual searches. Based on the implementations reported in the literature, differences are mainly at the level of the information stored in the memory and its use.

As indicated in the previous section, adaptive memory implementations follow the general principles of the sequential adaptive memory strategy (Rochat and Taillard 1995) and store the elements of the best solutions found by the individual threads together with, eventually, memories counting the frequency of each element in the best solutions encountered so far. Processes communicate their best solutions at the normal end of their search. New starting solutions are built out of the elements in the memory. When frequency memories are implemented, they are used to bias the selection of the elements during the construction phase.

The central memory approach has less limitations. In theory at least since, in our opinion, the potential of this method has not been fully explored yet. As far as we can tell, Crainic, Toulouse, and Gendreau (1997) proposed the first

central memory strategy for tabu search. A more fully developed implementation is reported by Crainic and Gendreau (2002) for the fixed cost, capacitated, multicommodity network design problem. In both studies, the individual tabu search threads communicate the vector of design variables each time the local solution is improved, but import a solution from the central memory only before undertaking a diversification phase. Then, if the imported solution is better than the current best, the diversification proceeds from the new solution. Otherwise, the imported solution is discarded and the procedure proceeds as usual. Memories are never re-initialized. The authors compared five strategies of retrieving a solution from the pool when requested by an individual thread. The strategy that always returns the overall best solution displayed the best performance when few (4) processors were used. When the number of processors was increased, a probabilistic procedure, based on the rank of the solution in the pool, appears to offer the best performance. The parallel procedure improves the quality of the solution and also requires less (wall clock) computing time compared to the sequential version, particularly for large problems with many commodities.

Recently, Le Bouthiller and Crainic (2004) took this approach one step further and proposed a central memory parallel meta-heuristic for the VRP with time windows where several tabu search and genetic algorithm threads co-operate. In this model, the central memory constitutes the population common to all genetic threads. Each genetic algorithm has its own parent selection and crossover operators. The offspring are returned to the pool to be enhanced by a tabu search procedure. The tabu search threads and the central memory follow the same rules as in the work of Crainic and Gendreau (2002). Experimental results show that without any particular calibration, the parallel meta-heuristic obtains solutions whose quality is comparable to the best meta-heuristics available, and demonstrates almost linear speedups. These results indicate that central memory approaches apply equally well whether complex constraint structures complement the combinatorial characteristics of the problems studied or not.

Many enhancements could be added to these co-operation mechanisms and should be studied further. One such enhancement, proposed but not yet tested in the literature, concerns performance measures that could be attached to each individual search. Then, processes that do not contribute significantly neither to the quality of the solutions in the pool, nor to their diversity could be discarded. The computing resources thus recuperated could either be simply released or assigned to more productive duties (e.g., speed up computations of some critical search steps or initiate new search threads modeled on the most successful ones). A second possibility concerns performance statistics attached to individual solutions (or parts thereof) measuring, for example, the relative improvement of solutions generated by individual threads when starting from them. Another open question concerns the global search memories that could

be inferred from the records built by individual searches and how the global search trajectory could then be inflected to yield superior results.

It has been noted that the pool of solutions constitutes a *population*. Then, other than the usual construction heuristics, population-based methods may be used to generate new solutions and improve the pool. Crainic and Gendreau (1999) report the development of such a hybrid search strategy combining their co-operative multi-thread parallel tabu search method with a genetic engine. The genetic algorithm initiates its population with the first elements from the central memory of the parallel tabu search. Asynchronous migration (migration rate = 1) subsequently transfers the best solution of the genetic pool to the parallel tabu search central memory, as well as solutions of the central memory towards the genetic population. Even though the strategy is not very elaborate, the hybrid appears to perform well, especially on larger problems. It is noteworthy that the genetic algorithm alone was not performing well and that it was the parallel tabu search procedure that identified the best results once the genetic method contributed to the quality of the central memory.

This combination of individual tabu search threads and population-based methods that run on the set of solutions in the central memory appears very promising and should be investigated in depth. (In more general terms, this strategy applies equally well to many meta-heuristic classes and is the object of research in the genetic and simulated annealing communities.) Different population-based methods could be used, *path relinking* and *scatter search* (Glover 1997, Glover and Laguna 1997, Glover, Laguna, and Martí 2000, Laguna and Martí 2003) being prime candidates. Both methods have proved highly successful on a wide spectrum of difficult problems. Moreover, the close relationships displayed by the two methods would facilitate the exchanges with co-operating tabu search threads. This topic must be thoroughly investigated.

This brings forth the issue of the parallelization of path relinking and scatter search. In their sequential versions, both methods make use of an initialization phase to generate “good” solutions, as well as of a *reference* or *elite* set of solutions continuously updated during the search. A path relinking method will select two solutions, a starting solution and a target one, and use a tabu search to move from the starting to the target solution. Moves are biased to favor gradually entering “good” attributes of the target solution into the current one. Scatter search, selects a number of solutions from the reference set and combines them to yield a new solution that may be improved by a local or tabu search algorithm. Solutions with “good” attributes, including overall best ones, are included by each method in the corresponding reference set.

Research on parallel path relinking and parallel scatter search is very scarce, yet. Of course, the strategies already presented for tabu search may be applied to path relinking and scatter search. Low level, 1-control, strategies, as well as multi-search approaches involving several path relinking or scatter search

threads, are straightforward. Co-operative strategies offer a greater challenge. Indeed, the reference set corresponds to a central memory co-operation mechanism that may be used by several path relinking or scatter search threads. Issues to be studied concern the initialization of the reference set and of each thread, as well as the particular definition of each thread. Thus, for example, one could consider the main algorithmic parts of scatter search – the selection of candidates, the generation of new solutions (different searches could combine a different number of solutions), and their possible improvement by a tabu search procedure – as “separate” procedures. A large number of co-operation designs become possible and should be investigated thoroughly. Also worthy of serious research is the role of memories in co-operation settings involving path relinking and scatter search methods, as well as the strategies that have the two methods co-operate through a unique reference set.

4. Perspectives and Research Directions

We have presented strategies to design parallel tabu search algorithms and surveyed developments and results in the area. Parallel tabu search methods offer the possibility to address problems more efficiently, both in terms of computing efficiency and solution quality. Low-level parallelization strategies appear often beneficial to speed up computation-intensive tasks, such as the evaluation of potential moves in the neighbourhood of a given solution. Moreover, such strategies may be advantageously incorporated into hierarchical parallel schemes where the higher level method either explores partitions of the solution domain or implements a co-operating multi-thread search.

Co-operation and multi-thread parallelization appear to offer the most interesting perspectives for tabu search, as for meta-heuristics in general. Asynchronous co-operative multi-thread strategies constitute probably the strongest trend in parallel meta-heuristics, the results reported in the literature indicating that they offer better results than synchronous and independent searches. They also seem to offer the most interesting development avenue for advanced tabu search methods, including path relinking and scatter search. More theoretical and empirical work is still required in this field, however. Consequently, in the second part of the paper, we focussed on multi-search strategies. We discussed general design and implementation principles, pointed out a number of challenges and pitfalls, and identified trends and promising research directions.

Hybridization is another important trend in sequential and parallel meta-heuristics and recent studies tend to demonstrate that combining different meta-heuristics yields superior results. This opens up an exciting field of enquiry. What meta-heuristics to combine with tabu search? What role can each type of meta-heuristic play? What information is exchanged and how it is used in this context? How important a role path relinking and scatter search may play for

parallel meta-heuristics in general? These are only a few of the questions that need to be explored.

It has been often noticed that co-operating parallel mechanisms bear little, if any, resemblance to the initial meta-heuristic one attempts to parallelize. This remark is true whether the individual search threads belong to the same class of meta-heuristics or not. A natural question then is whether co-operating multi-thread parallel methods should form a “new”, very broadly defined, class of meta-heuristics.

Is co-operative multi-thread parallelization a “new” meta-heuristic? Almost certainly, even though we face the challenge to properly define it. Is it a “new” tabu search class? Not as currently used. Indeed, the method misses a global search control mechanism and, especially, the memories that would guide this mechanism are a central component of tabu search. The development of such memories and mechanisms is one of the most interesting challenges we face.

Acknowledgments

Funding for this project has been provided by the Natural Sciences and Engineering Council of Canada and by the Fonds F.C.A.R. of the Province of Québec. While working on the project, Dr. T.G. Crainic was Adjunct Professor at the Département d’Informatique et de Recherche Opérationnelle of the Université de Montréal (Canada).

References

- Aiex, R.M., S.L. Martins, C.C. Ribeiro and N.R. Rodriguez (1996) Asynchronous Parallel Strategies for Tabu Search Applied to the Partitioning of VLSI Circuits, Monografias em ciência da computação, Pontifícia Universidade Católica de Rio de Janeiro.
- Aiex, R.M., S.L. Martins, C.C. Ribeiro and N.R. Rodriguez (1998) "Cooperative Multi-Thread Parallel Tabu Search with an Application to Circuit Partitioning," In *Proceedings of IRREGULAR'98 - 5th International Symposium on Solving Irregularly Structured Problems in Parallel, Lecture Notes in Computer Science*, Springer-Verlag, 1457:310–331.
- Andreatta, A.A. and C.C. Ribeiro (1994) "A Graph Partitioning Heuristic for the Parallel Pseudo-Exhaustive Logical Test of VLSI Combinational Circuits," *Annals of Operations Research*, 50:1–36.
- Badeau, P., F. Guertin, M. Gendreau, J.Y. Potvin and É.D. Taillard (1997) "A Parallel Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows," *Transportation Research C: Emerging Technologies*, 5(2):109–122.

- Barr, R.S. and B.L. Hickman (1993) "Reporting Computational Experiments with Parallel Algorithms: Issues, Measures, and Experts Opinions," *ORSA Journal on Computing*, 5(1):2–18.
- Battiti, R. and G. Tecchiolli (1992) "Parallel Based Search for Combinatorial Optimization: Genetic Algorithms and TABU," *Microprocessors and Microsystems*, 16(7):351–367.
- Chakrapani, J. and J. Skorin-Kapov (1992) "A Connectionist Approach to the Quadratic Assignment Problem," *Computers & Operations Research*, 19(3/4):287–295.
- Chakrapani, J. and J. Skorin-Kapov (1993) "Connection Machine Implementation of a Tabu Search Algorithm for the Traveling Salesman Problem," *Journal of Computing and Information Technology*, 1(1):29–36.
- Chakrapani, J. and J. Skorin-Kapov (1993). "Massively Parallel Tabu Search for the Quadratic Assignment Problem," *Annals of Operations Research*, 41:327–341.
- Chakrapani, J. and J. Skorin-Kapov (1995) "Mapping Tasks to Processors to Minimize Communication Time in a Multiprocessor System," In *The Impact of Emerging Technologies of Computer Science and Operations Research*, Kluwer Academic Publishers, Norwell, MA, 45–64.
- Crainic, T.G. and M. Gendreau (1999) "Towards an Evolutionary Method - Cooperating Multi-Thread Parallel Tabu Search Hybrid," In S. Voß, S. Martello, C. Roucairol and I.H. Osman, Editors, *Meta-Heuristics 98: Theory & Applications*, Kluwer Academic Publishers, Norwell, MA, 331–344.
- Crainic, T.G. and M. Gendreau (2002) "Cooperative Parallel Tabu Search for Capacitated Network Design," *Journal of Heuristics*, 8(6):601–627.
- Crainic, T.G. and M. Toulouse (1998) *Parallel Metaheuristics*, In T.G. Crainic and G. Laporte, Editors, *Fleet Management and Logistics*, Kluwer Academic Publishers, Norwell, MA, 205–251.
- Crainic, T.G. and M. Toulouse (2003) *Parallel Strategies for Meta-heuristics*, In F. Glover and G. Kochenberger, Editors, *State-of-the-Art Handbook in Metaheuristics*, Kluwer Academic Publishers, Norwell, MA, 475–513.
- Crainic, T.G., M. Toulouse and M. Gendreau (1995) "Parallel Asynchronous Tabu Search for Multicommodity Location-Allocation with Balancing Requirements," *Annals of Operations Research*, 63:277–299.
- Crainic, T.G., M. Toulouse and M. Gendreau (1995) "Synchronous Tabu Search Parallelization Strategies for Multicommodity Location-Allocation with Balancing Requirements," *OR Spektrum*, 17(2/3):113–123.
- Crainic, T.G., M. Toulouse and M. Gendreau (1997) "Towards a Taxonomy of Parallel Tabu Search Algorithms," *INFORMS Journal on Computing*, 9(1):61–72.
- Cung, V.D., S.L. Martins, C.C. Ribeiro and C. Roucairol (2002) "Strategies for the Parallel Implementations of Metaheuristics," In C. Ribeiro and P.

- Hansen, Editors, *Essays and Surveys in Metaheuristics*, Kluwer Academic Publishers, Norwell, MA, 263–308.
- De Falco, I., R. Del Balio and E. Tarantino (1995) Solving the Mapping Problem by Parallel Tabu Search. Report, Istituto per la Ricerca sui Sistemi Informatici Paralleli-CNR.
- De Falco, I., R. Del Balio, E. Tarantino and R. Vaccaro (1994) "Improving Search by Incorporating Evolution Principles in Parallel Tabu Search," In *Proceedings International Conference on Machine Learning*, 823–828.
- Fiechter, C.N. (1994) "A Parallel Tabu Search Algorithm for Large Travelling Salesman Problems," *Discrete Applied Mathematics*, 51(3):243–267.
- Garcia, B.L., J.Y. Potvin and J.M. Rousseau (1994) "A Parallel Implementation of the Tabu Search Heuristic for Vehicle Routing Problems with Time Window Constraints," *Computers & Operations Research*, 21(9):1025–1033.
- Gendreau, M., F. Guertin, J.Y. Potvin and É.D. Taillard (1999) "Tabu Search for Real-Time Vehicle Routing and Dispatching," *Transportation Science*, 33(4):381–390.
- Glover, F. (1986) "Future Paths for Integer Programming and Links to Artificial Intelligence," *Computers & Operations Research*, 1(3):533–549.
- Glover, F. (1989) "Tabu Search – Part I," *ORSA Journal on Computing*, 1(3):190–206.
- Glover, F. (1990) "Tabu Search – Part II," *ORSA Journal on Computing*, 2(1):4–32.
- Glover, F. (1996) "Tabu Search and Adaptive Memory Programming – Advances, Applications and Challenges," In R. Barr, R. Helgason and J. Kennington, Editors, *Interfaces in Computer Science and Operations Research*, Kluwer Academic Publishers, Norwell, MA, 1–75.
- Glover, F. (1997) "A Template for Scatter Search and Path Relinking," In J. Hao, E. Lutton, E. Ronald, M. Schoenauer and D. Snyers, Editors, *Artificial Evolution, Lecture Notes in Computer Science*, Springer Verlag, Berlin, 1363:13–54.
- Glover, F. and M. Laguna (1993) "Tabu Search," In C. Reeves, Editor, *Modern Heuristic Techniques for Combinatorial Problems*, Blackwell Scientific Publications, Oxford, 70–150.
- Glover, F. and M. Laguna (1997) *Tabu Search*. Kluwer Academic Publishers, Norwell, MA.
- Glover, F., M. Laguna and R. Martí (2000) "Fundamentals of Scatter Search and Path Relinking," *Control and Cybernetics*, 39(3):653–684.
- Glover, F., É.D. Taillard and D. de Werra (1993) "A User's Guide to Tabu Search," *Annals of Operations Research*, 41:3–28.
- Holmqvist, K., A. Migdalas and P.M. Pardalos (1997) "Parallelized Heuristics for Combinatorial Search," In A. Migdalas, P. Pardalos and S. Stroy, Editors,

- Parallel Computing in Optimization*, Kluwer Academic Publishers, Norwell, MA, 269–294.
- Laguna, M. and R. Martí (2003) *Scatter Search: Methodology and IMplementations in C*. Kluwer Academic Publishers, Norwell, MA.
- Le Bouthillier, A. and T.G. Crainic (2004) "A Cooperative Parallel Meta-Heuristic for the Vehicle Routing Problem with Time Windows," *Computers & Operations Research*.
- Malek, M., M. Guruswamy, M. Pandya and H. Owens (1989) "Serial and Parallel Simulated Annealing and Tabu Search Algorithms for the Traveling Salesman Problem," *Annals of Operations Research*, 21:59–84.
- Martins, S.L., C.C. Ribeiro and N.R. Rodriguez (1996) *Parallel Programming Tools for Distributed Memory Environments*, Monografias em Ciência da Computação, Pontifícia Universidade Católica de Rio de Janeiro.
- Ouyang, M., M. Toulouse, K. Thulasiraman, F. Glover and J.S. Deogun (2000) "Multi-Level Cooperative Search: Application to the Netlist/Hypergraph Partitioning Problem," In *Proceedings of International Symposium on Physical Design*, ACM Press, 192–198.
- Ouyang, M., M. Toulouse, K. Thulasiraman, F. Glover and J.S. Deogun (2002) "Multilevel Cooperative Search for the Circuit/Hypergraph Partitioning Problem," *IEEE Transactions on Computer-Aided Design*, 21(6):685–693.
- Pardalos, P.M., L. Pitsoulis, T. Mavridou and M.G.C. Resende (1995) "Parallel Search for Combinatorial Optimization: Genetic Algorithms, Simulated Annealing, Tabu Search and GRASP," In A. Ferreira and J. Rolim, Editors, *Proceedings of Workshop on Parallel Algorithms for Irregularly Structured Problems, Lecture Notes in Computer Science 980*, Springer-Verlag, Berlin, 317–331.
- Porto, S.C.S., J.P.F.W. Kitajima and C.C. Ribeiro (2000) "Performance Evaluation of a Parallel Tabu Search Task Scheduling Algorithm," *Parallel Computing*, 26:73–90.
- Porto, S.C.S. and C.C. Ribeiro (1995) "A Tabu Search Approach to Task Scheduling on Heterogeneous Processors Under Precedence Constraints," *International Journal of High-Speed Computing*, 7:45–71.
- Porto, S.C.S. and C.C. Ribeiro (1996) "Parallel Tabu Search Message-Passing Synchronous Strategies for Task Scheduling Under Precedence Constraints," *Journal of Heuristics*, 1(2):207–223.
- Rego, C. and C. Roucairol (1996) "A Parallel Tabu Search Algorithm Using Ejection Chains for the VRP," In I. Osman, and J. Kelly, Editors, *Meta-Heuristics: Theory & Applications*, Kluwer Academic Publishers, Norwell, MA, 253–295.
- Rochat, Y. and É.D. Taillard (1995) "Probabilistic Diversification and Intensification in Local Search for Vehicle Routing," *Journal of Heuristics*, 1(1):147–167.

- Schulze, J. and T. Fahle (1999) "A Parallel Algorithm for the Vehicle Routing Problem with Time Window Constraints," *Annals of Operations Research*, 86:585–607.
- Taillard, É.D. (1991) "Robust Taboo Search for the Quadratic Assignment Problem," *Parallel Computing*, 17:443–455.
- Taillard, É.D. (1993) "Parallel Iterative Search Methods for Vehicle Routing Problems," *Networks*, 23:661–673.
- Taillard, É.D. (1993) *Recherches itératives dirigées parallèles*. PhD thesis, École Polytechnique Fédérale de Lausanne.
- Taillard, É.D. (1994) "Parallel Taboo Search Techniques for the Job Shop Scheduling Problem," *ORSA Journal on Computing*, 6(2):108–117.
- Taillard, É.D., P. Badeau, M. Gendreau, F. Guertin and J.Y. Potvin (1997) "A Tabu Search Heuristic for the Vehicle Routing Problem with Soft Time Windows," *Transportation Science*, 31(2):170–186.
- Toulouse, M., T.G. Crainic and M. Gendreau (1996) "Communication Issues in Designing Cooperative Multi Thread Parallel Searches," In I.H. Osman and J.P. Kelly, Editors, *Meta-Heuristics: Theory & Applications*, Kluwer Academic Publishers, Norwell, MA, 501–522.
- Toulouse, M., T.G. Crainic and B. Sansó (1999) "An Experimental Study of Systemic Behavior of Cooperative Search Algorithms," In S. Voß, S. Martello, C. Roucairol and I.H. Osman, Editors, *Meta-Heuristics 98: Theory & Applications*, Kluwer Academic Publishers, Norwell, MA, 373–392.
- Toulouse, M., T.G. Crainic and B. Sansó (2004) "Systemic Behavior of Cooperative Search Algorithms," *Parallel Computing*, 21(1):57–79.
- Toulouse, M., T.G. Crainic, B. Sansó and K. Thulasiraman (1998). "Self-Organization in Cooperative Search Algorithms," In *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics*, Omnipress, Madison, Wisconsin, 2379–2385.
- Toulouse, M., T.G. Crainic and K. Thulasiraman (2000) "Global Optimization Properties of Parallel Cooperative Search Algorithms: A Simulation Study," *Parallel Computing*, 26(1):91–112.
- Toulouse, M., F. Glover and K. Thulasiraman (1998) "A Multi-Scale Cooperative Search with an Application to Graph Partitioning," Report, School of Computer Science, University of Oklahoma, Norman, OK.
- Toulouse, M., K. Thulasiraman and F. Glover (1999). "Multi-Level Cooperative Search," In P. Amestoy, P. Berger, M. Daydé, I. Duff, V. Frayssé, L. Giraud and D. Ruiz, Editors, *5th International Euro-Par Parallel Processing Conference, Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1685:533–542.
- Verhoeven, M.G.A. and E.H.L. Aarts (1995) "Parallel Local Search," *Journal of Heuristics*, 1(1):43–65.

Voß, S. (1993) "Tabu Search: Applications and Prospects," In D.Z. Du and P. Pardalos, Editors, *Network Optimization Problems*, World Scientific Publishing Co., Singapore, 333–353.