CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Parallel Meta-Heuristic Search

Teodor Gabriel Crainic

August 2015

CIRRELT-2015-42

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Parallel Meta-Heuristic Search

Teodor Gabriel Crainic*

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Management and Technology, Université du Québec à Montréal, P.O. Box 8888, Station Centre-Ville, Montréal, Canada H3C 3P8

Abstract. We present a general picture of the parallel meta-heuristic search for optimization. We recall the main concepts and strategies in designing parallel meta-heuristics, pointing to a number of contributions that instantiated them for neighborhood- and population- based meta-heuristics, and identify trends and promising research directions. We focus on cooperation-based strategies, which display remarkable performances, in particular on asynchronous cooperation and advanced cooperation strategies that create new information out of exchanged data to enhance the global guidance of the search.

Keywords: Parallel computation, parallel meta-heuristics, functional and data decomposition, independent multi-search, synchronous and asynchronous cooperative search, integrative cooperative search.

Acknowledgements. The author wishes to acknowledge the contributions of his colleagues and students, in particular Professors Michel Gendreau, Université de Montréal, Canada, and Michel Toulouse, the Vietnamese-German University, Vietnam, who collaborated over the years to the work on parallel meta-heuristics for combinatorial optimization. All errors are solely and entirely due to the author, however. Partial funding for this project has been provided by the Natural Sciences and Engineering Research Council of Canada (NSERC), through its Discovery Grant and the Discovery Accelerator Supplements programs, and the Strategic Clusters program of the Fonds de recherche du Québec – Nature et technologie (FRQNT). The author thanks the two Institutions for supporting this research.

---

* Corresponding author: TeodorGabriel.Crainic@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2015

1 Introduction

Meta-heuristics often offer the only practical approach to addressing complex problems of realistic dimensions, and are thus widely acknowledged as essential tools in numerous and diverse fields. Yet, even meta-heuristics may reach quite rapidly the limits of what may be addressed in acceptable computing times for many problem settings for research and practice alike. Moreover, heuristics do not generally guaranty optimality, performance often depending on the particular problem setting and instance characteristics. *Robustness* is therefore a major objective in meta-heuristic design, in the sense of offering a consistently high level of performance over a wide variety of problem settings and instance characteristics.

Parallel meta-heuristics aim to address both issues. Their first goal is to solve larger problem instances than what is achievable by sequential methods, and to do it in reasonable computing times. In appropriate settings, such as cooperative multi-search strategies, parallel meta-heuristics also prove to be much more robust than sequential versions in dealing with differences in problem types and characteristics. They also require less extensive, and expensive, parameter-calibration efforts.

The objective of this chapter is to paint a general picture of the parallel optimization meta-heuristic field. Following Crainic and Toulouse (2010), we recall the main concepts and strategies in designing parallel meta-heuristics, pointing to a number of contributions that instantiated them for neighborhood- and population-based meta-heuristics, and identify a number of trends and promising research directions. We focus in particular on cooperation-based strategies, which display remarkable performances, reviewing the recently-introduced Integrative Cooperative Search (Lahrichi et al., 2015). Notice that we examine and discuss the strategies from the conceptual, algorithmic-design point of view, independent of implementation on particular computer architectures.

The parallel meta-heuristic field is very broad, while the space available for this chapter is limited. In addition to the references provided in the following sections, the reader may consult a number of surveys, taxonomies, and syntheses of parallel meta-heuristics, some addressing methods based on particular methodologies, while others address the field in more comprehensive terms. Methodology-dedicated syntheses may be found in Azencott (1992); Greening (1989, 1990b,a); Ram et al. (1996) for parallel simulated annealing, Cantú-Paz (1998, 2005); Lin et al. (1994); Mühlenbein (1992); Shonkwiler (1993) for genetic-based evolutionary methods, Crainic (2005); Crainic et al. (2005, 1997); Glover and Laguna (1997); Voß (1993) for tabu search; García-López et al. (2005) for scatter search, Bullnheimer et al. (1999); Dorigo and Stuetzle (2003); Janson et al. (2005) for ant-colony methods, and Moreno-Pérez et al. (2005) for Variable Neighborhood Search (VNS). Surveys and syntheses that address more than one methodology may be found in Alba (2005); Alba et al. (2013); Crainic (2008); Crainic and Hail (2005); Crainic and Toulouse (1998, 2003, 2010); Crainic et al. (2014); Cung et al. (2002); Holmqvist et al.

(1997); Laursen (1996); Pardalos et al. (1995a); Verhoeven and Aarts (1995).

The chapter is organized as follows. Section 2 *Meta-heuristics and Parallelism* is dedicated to a general discussion of the potential for parallel computing in meta-heuristics, a brief description of performance indicators for parallel meta-heuristics, and the taxonomy we use to structure the presentation. Section 3 *Low-Level Parallelization Strategies* addresses strategies focusing on accelerating computing-intensive tasks without modifying the basic algorithmic design. Methods based on the decomposition of the search space are treated in Section 4 *Data Decomposition*, while strategies based on the simultaneous exploration of the search space by several independent meta-heuristics constitutes the topic of Section 5 *Independent Multi-search*. Cooperation principles are discussed in Section 6 *Cooperative Search* and are detailed in Sections 6.1 *Synchronous Cooperation*, 6.2 *Asynchronous Cooperation*, and 7 *Advanced Cooperation Strategies - Creating Knowledge*. We conclude in Section 8 *Perspectives* .

2 Meta-heuristics and Parallelism

We start with a brief overview of the potential for parallel computing in meta-heuristics and of performance indicators for parallel meta-heuristics. We conclude with the criteria used to describe and characterize the parallelization strategies for meta-heuristics.

2.1 Sources of Parallelism

Addressing a given problem instance with a parallel solution method means that several processes work simultaneously on several processors with the goal of identifying the best solution for the instance. Parallelism thus follows from a decomposition of the total work load and the distribution of the resulting tasks to available processors. According to how " small" or "large" are the tasks in terms of algorithm work or search space, the parallelization is denoted *fine-* or *coarse-grained*, respectively.

The decomposition may concern the algorithm, the problem-instance data, or the problem structure. *Functional parallelism* identifies the first case, where some computing-intensive components of the algorithm are separated into several tasks (processes), possibly working on the "same" data, which are allocated to different processors and run in parallel. The main source of functional parallelism for meta-heuristics is the concurrent execution of their innermost loop iterations, e.g., evaluating neighbors, computing the fitness of individuals, or having ants forage concurrently (Section *Low-Level Parallelization Strategies*). This is often also the only source of readily available parallelism in meta-heuristics, most other steps being time dependent and requiring either the computation

of the previous steps to be completed, or the synchronization of computations to enforce this time-dependency. Consequently, functional parallelism is mainly interesting as a low-level component of hierarchical parallelization strategies or when addressing problem settings requiring a significant part of the computing effort be spent in the inner-loop algorithmic component.

Parallelism for meta-heuristics may further be found in the domain of the problem addressed or the corresponding search space (for brevity reasons, the term "search space" is used in the rest of the chapter). There are indeed no data dependencies between the evaluation functions of different solutions and, thus, these may be computed in parallel. Moreover, theoretically, the parallelism in the search space is as large as the space itself. Parallelism is then obtained by separating the search space into components allocated to the available processors. In most cases, these components are still too large for explicit enumeration, and an exact or heuristic search method has to be associated to each to implicitly explore it. *Space separation* is exploited in most of the strategies described in this chapter.

Space separation raises a number of issues with respect to defining an overall meta-heuristic search strategy, in particular, 1) whether to define the separation by partitioning the space, allowing components to partially overlap, or not; 2) how to control an overall search conducted separately on several components of the original space; 3) how to build a complete solution out of those obtained while exploring the components; 4) how to allocate computing resources for an efficient exploration avoiding, for example, searching regions with poor-quality solutions.

Two main approaches are used to perform the search-space separation: *domain decomposition* (also called *data parallelism*) and *multi search* (the name *multiple walks* is also found in the literature). The former explicitly separates the search space (Section 4) *Data Decomposition* , and then addresses the initial problem instantiated on each of the resulting restricted regions, before combining the corresponding partial solutions into complete ones.

Multi-search strategies implicitly divide the search space through concurrent explorations by several methods, named *solvers* in the following. These meta-heuristic or exact solvers may address either the complete problem at hand, or explore partial problems defined by decomposing the initial problem through mathematical programming or attribute-based heuristic approaches. In the former case, the decomposition method implicitly defines how a complete solution is built out of partial ones. In the latter case, some processors work on the partial problems corresponding to the particular sets of attributes defined in the decomposition, while others combine the resulting partial solutions into complete solutions to the original problem. Multi-search strategies, particularly those based on cooperation principles, make up the bulk of the successful parallel meta-heuristics developments, and are the object of most recent publications in the

field. We discuss them in Sections 5 *Independent Multi-search*, 6 *Cooperative Search*, 6.1 *Synchronous Cooperation*, 6.2 *Asynchronous Cooperation*, and 7 *Advanced Cooperation Strategies - Creating Knowledge*.

We complete this subsection with a few notes on the performance evaluation of parallel meta-heuristics strategies and resulting algorithms.

The traditional goal when designing parallel solution methods is to reduce the time required to "solve", exactly or heuristically, given problem instances or to address larger instances without increasing the computational effort. For solution methods that run until the provable optimal solution is obtained, this translates into the well-known *speedup* performance measure, computed as the ratio between the wall-clock time required to solve the problem instance in parallel with p processors and the corresponding solution time of the best-known sequential algorithm; A somewhat less restrictive measure replaces the latter with the time of the parallel algorithm run on a single processor. See Barr and Hickman (1993) for a detailed discussion of this issue, including additional performance measures.

Speedup measures are more difficult to interpret when the optimal solution is not guaranteed or the exact method is stopped before optimality is reached. Moreover, most parallelization strategies design parallel meta-heuristics that yield solutions that are different in value, composition, or both, from those obtained by the sequential counterpart. Thus, equally important measures for parallel heuristics are by how much they outperform their sequential counterparts in (relative) terms of solution quality and, ideally, computational efficiency. Simply put, the parallel method should not require a higher overall computation effort than the sequential method or should justify the extra effort by higher quality solutions.

Search robustness is another characteristic expected of parallel heuristics. Robustness with respect to a problem setting is meant here in the sense of providing "equally" good solutions to a large and varied set of problem instances, without excessive calibration, neither during the initial development, nor when addressing new problem instances.

Multi-search methods, particularly those based on cooperation, generally offer enhanced performances compared to sequential methods and other parallelization strategies. They display behaviors different from those of the sequential methods involved and can be seen as proper meta-heuristics (Alba, 2005), usually finding better-quality solutions and enhancing the robustness of the meta-heuristic search. See Crainic and Toulouse (1998, 2003) for a discussion of these issues.

2.2 Characterizing Parallel Meta-heuristic Strategies

Several strategies may be defined based on each one of the sources of parallelism discussed above. We adopt the classification of Crainic and Hail (2005), generalizing that of Crainic et al. (1997) (Verhoeven and Aarts (1995) and Cung et al. (2002) present classifications that proceed of the same spirit), to characterize these strategies.

The three dimensions of the classification focus on the control of the global problem-solving process, the information exchanged among processes, and the diversity of the search, respectively. The first dimension, *Search Control Cardinality*, thus specifies whether the global search is controlled by a single process or by several processes that may collaborate or not. The two alternatives are identified as *1-control* (*1C*) and *p-control* (*pC*), respectively.

The second dimension addresses the issue of information exchanges and the utilization of the exchanged information to control or guide the search; hence the *Search Control and Communications* name. Communications may proceed either *synchronously* or *asynchronously*. In the former case, processes stop and engage in some form of communication and information exchange at moments (number of iterations, time intervals, specified algorithmic stages, etc.) exogenously planned, either hard-coded or determined by a control (master) process. In the asynchronous communication case, each process is in charge of its own search, as well as of establishing communications with other processes, and the global search terminates once all individual searches stop. To reflect the quantity and quality of the information exchanged and shared, as well as the additional knowledge derived from these exchanges (if any), these notions conveyed through four strategy classes: *Rigid (RS)* and *Knowledge Synchronization (KS)* and, symmetrically, *Collegial (C)* and *Knowledge Collegial (KC)*.

More than one solution method or variant (e.g., with different parameter settings) may be involved in a parallel meta-heuristic. The third dimension thus indicates the *Search Differentiation* or diversity: do solvers start from the same or different solutions and do they make use of the same or different search strategies? The four classes are: *SPSS, Same initial Point/Population, Same search Strategy*; *SPDS, Same initial Point/Population, Different search Strategies*; *MPSS, Multiple initial Points/Populations, Same search Strategies*; *MPDS, Multiple initial Points/Populations, Different search Strategies*. Obviously, one uses "point" for neighborhood-based methods, while "population" is used for genetic-based evolutionary methods, scatter search, and swarm methods.

3 Low-Level Parallelization Strategies

Functional-parallelism-based strategies, exploiting the potential for task decomposition within the inner-loop computations of meta-heuristics, are often labeled "low level" because they modify neither the algorithmic logic, nor the search space. They aim solely to accelerate the search and generally do not modify the search behavior of the sequential meta-heuristic. Typically, the exploration is initialized from a single initial solution or population, and proceeds according to the sequential meta-heuristic logic, while a number of intensive-computation steps are decomposed and simultaneously performed by several processors.

Most low-level parallel strategies belong to the 1C/RS/SPSS class and are usually implemented according to the classical *master-slave* parallel programming model. A "master" program executes the 1-control sequential meta-heuristic, separating and dispatching computation-intensive tasks to be executed in parallel by "slave" programs. Slaves perform evaluations and return the results to the master which, once all the results are in, resumes the normal logic of the sequential meta-heuristic. The complete control on the algorithm execution thus rests with the master, which decides the work allocation for all other processors and initiates most communications. No communications take place among slave programs.

The neighborhood-evaluation procedure of the Local Search component of neighborhood-based meta-heuristics (as well as of population-based ones implementing advanced "schooling" for offspring) is generally targeted in 1C/RS/SPSS designs. The master groups the neighbors into tasks and sends them to slaves. Each slave then executes the exploration/evaluation procedure on its respective part of the neighborhood and sends back the best, or first improving, neighbor found. The master waits for all slaves to terminate their computations, selects the best move and proceeds with the search. See, e.g., Garcia et al. (1994) and Porto and Ribeiro (1996) for applications of this strategy to tabu search meta-heuristics for the vehicle routing problem with time-window constraints (VRPTW) and the scheduling dependent tasks on heterogeneous processors, respectively.

The appropriate granularity of the decomposition, that is the size of the tasks, depends upon the particular problem and computer architecture, but may generally be computationally sensitive to inter-processor communication times and work-load balancing. Thus, for example, Davidović and Crainic (2015) discusses several decomposition policies for the permutation-based Local Search neighborhood applied to scheduling dependent tasks on homogeneous processors and show that the uniform partition usually called upon in the literature is not appropriate in this context characterized by neighborhoods of different sizes. They also show that a fixed coarse-grained non-uniform decomposition, while offering superior results, requires calibration each time the problem size or the number of processors varies.

The best performing strategy was called by the authors *dynamic fine-grained*. It defines each neighbor evaluation as a single task, the master dynamically dispatching these on a first-available, first-served basis to slave processors as they complete their tasks. The strategy partitions the neighborhood into a number of components equal to the number of available processors, but of unequal size with a content dynamically determined at each iteration.

The dynamic fine-grained strategy provides maximum flexibility and good load balancing, particularly when the evaluation of neighbors is of uneven length. The uniform distribution appears more appropriate when the neighbor evaluations are sensibly the same, or when the overhead cost of the dynamic strategy for creating and exchanging tasks appears too high.

Similar observations may be made regarding population-based meta-heuristics. In theory, all genetic-algorithm operators may be addressed through a 1C/RS/SPSS design, and the degree on possible parallelism is equal to the population size. In practice, the computations associated to most operators are not sufficiently heavy to warrant parallelization, while overhead costs may significantly reduce the degree of parallelism and increase the granularity of the tasks. Consequently, the fitness evaluation is often the only target of 1C/RS/SPSS parallelism for genetic-evolutionary methods, the resulting parallel GA being implemented using the master-slave model. Similarly to other 1-control low-level parallelizations, a 1C/RS/SPSS evolutionary-genetic algorithm performs the same search as the sequential program, only faster.

The 1C/RS/SPSS parallelism for ant-colony and, more generally, swarm-based methods lies at the level of the individual ants. Ants share information indirectly through the pheromone matrix, which is updated once all solutions have been constructed. There are no modifications of the pheromone matrix during a construction cycle and, thus, each individual ant performs its solution-construction procedure without data dependencies on the progress of the other ants.

Most parallel ant-colony methods implement some form of 1C/RS/SPSS strategy according to the master-slave model, including Bullnheimer et al. (1999); Doerner et al. (2004); Rahoual et al. (2002); Randall and Lewis (2002); Talbi et al. (1999). The master builds tasks consisting of one or several ants (each can be assimilated to a "small" colony) and distributes them to the available processors. Slaves perform their construction heuristic and return their solution(s) to the master, which updates the pheromone matrix, returns it to the slaves, and so on. To further speed up computation, the pheromone update can be partially computed at the level of each slave, each computing the update associated to its solutions. The fine-grained version with central matrix update has been the topic of most contributions so far and, in general, it outperformed the sequential version of the algorithm. It is acknowledged, however, that it does not scale and, similarly to other meta-heuristics, this strategy is outperformed by more advanced multi-search

methods.

Scatter search and path relinking implement different evolution strategies, where a restricted number of elite solutions are combined, the result being enhanced through a local search or a full-fledged meta-heuristic, usually neighborhood-based. Consequently, the 1C/RS/SPSS strategies discussed previously regarding the parallelization of local-search exploration apply straightforwardly to the present context, as in García-López et al. (2005, 2006, 2003) for the p-median and the feature-selection problems.

A different 1C/RS/SPSS strategy for scatter search may be obtained by running concurrently the combination and improvement operators on several subsets of the reference set. Here, the master generates tasks by extracting a number of solution subsets, which are sent to slaves. Each slave then combines and improves its solutions, returning its results to the master for the global update of the reference set. Each subset sent to a slave may contain exactly the number of solutions required by the combination operator or a higher number. In the former case García-López et al. (2005, 2006, 2003), the slave performs an "iteration" of the scatter search algorithm. In the latter, several combination-improvement sequences could be executed and solutions could be returned to the master as they are found or all together at the end of all sequences. This heavy load for slaves may conduct to very different computation times and, thus, load-balancing capabilities should be added to the master.

To conclude, low level, 1-control parallel strategies are particularly attractive when neighborhoods (populations) are large or neighbor (individual) evaluation procedures are costly and a significant gain in computing time may be obtained (e.g., the parallel tabu searches of Chakrapani and Skorin-Kapov (1992, 1993a); Taillard (1991) for the Quadratic Assignment Problem (QAP), Chakrapani and Skorin-Kapov (1993b) for the Traveling Salesman Problem (TSP), Porto and Ribeiro (1995, 1996); Porto et al. (2000) and Davidović and Crainic (2015) for the task-scheduling problem). More advanced multi-search strategies generally outperform low-level strategies in most cases. Yet, when a sufficiently large number of processors is available, it might prove worthy to combine a 1C/RS/SPSS approach and more sophisticated strategies into hierarchical solution schemes (e.g., Rego and Roucairol (1996) were low-level parallelism accelerated the move evaluations of the individual searches engaged into an independent multi-search procedure for the VRP).

4 Data Decomposition

Domain or *search-space decomposition* constitutes an intuitively simple and appealing parallelization strategy, dividing the search space into smaller partial sets, solving the resulting subproblems by applying the sequential meta-heuristic on each set, collecting

the respective partial solutions, and reconstructing an entire solution out of the partial ones. This apparently simple idea may take several forms, however, according to the type of division performed and the permitted links among the resulting set/subproblems.

The space may be *partitioned*, yielding disjoint partial sets, or a cover may be defined allowing a certain amount of overlap among partial sets. Thus, for example, the arc-design variables of a VRP may be partitioned into customer subsets (including the depot in each subset), while a cover would allow "close by" customers to belong to two subsets. The goal generally is to generate independent subproblems, which allows to discard from each subproblem the variables and constraints not directly involved in its definition. When this is not possible, e.g., the separated activities share some resources, one may fix the variables not in the subproblem definition (and thus project the corresponding constraints). One then still works on a smaller subproblem, but considering the complete vector of decision variables.

The second element one must consider is the degree of exploration overlap permitted among subproblems. One must thus decide whether the solution transformations (e.g., neighborhood moves or individual crossover) performed within the partial set of a given subproblem are restricted to that partial set, or may involve variables in neighboring subspaces creating an indirect overlapping of subsets. Strict partitioning strategies restrict the solvers to their subsets, which results in part of the search space being unreachable and the parallel meta-heuristic being non-optimal. Explicit or implicit overlapping partially addresses this issue. Only partially because, to fully ensure that all potential solutions are reachable, one needs to make overlapping cover the entire search space, which would negate the benefits of decomposition.

Consequently, strict partitioning or very limited overlapping are the preferred approaches found in the literature. A re-decomposition feature is generally included to increase the thoroughness of the search and allow all potential solutions to be examined. The decomposition is thus modified at regular intervals and the search is restarted using the new decomposition. This feature provides also the opportunity to define a non-exhaustive decomposition, i.e., where the union of the subsets is smaller than the complete search space. A complete-solution reconstruction feature is almost always part of the procedure.

This strategy is naturally implemented using master-slave 1C/RS schemes, with MPSS or MPDS search-differentiation. The master process determines the decomposition and sends partial sets to slaves, synchronizes them and collects their solutions, reconstructs solutions, modifies the decomposition, and determines stopping conditions. Slaves concurrently and independently perform the search on their assigned partial sets. Design issues one must address in this context are the length of the exploration available to slaves, and the reconstruction of global context information (e.g., global tabu list) out of the partial ones. The extreme case of executing a full meta-heuristic on each

partial set of the search space (this avoids the context issue), periodically modifying the partition and re-starting the search, was actually generally used, particularly for problems for which a large number of iterations can be performed in a relatively short time and restarting the method with a new decomposition does not require an unreasonable computational effort (e.g., Fiechter (1994) for the TSP, Laganière and Mitiche (1995) for image filtering, and Gendreau et al. (2001) for real-time ambulance fleet management).

In a pC/KS strategy, with MPSS or MPDS search-differentiation, the decomposition is collegially decided and modified through information exchange phases (e.g., round-robin or many-to-many exchanges) activated at given synchronization points. Such an approach was proposed in Taillard (1993) for the VRP, where the customer set was partitioned, vehicles were allocated to the resulting regions, and each subproblem was solved by an independent tabu search. All processors stopped after a number of iterations that varied according to the total number of iterations already performed, and the partition was modified by exchanging tours, undelivered cities, and empty vehicles between adjacent processors (corresponding to neighboring regions). At the time, this approach did allow to address successfully a number of problem instances, but the synchronization inherent in the design of the strategy hindered its performance. A parallel ant-colony approach combining this decomposition idea to a master-slave implementation was presented in Doerner et al. (2005) (parallelizing the algorithm presented in Reimann et al. (2004)), where the master generates an initial solution, defines the partition, and updates the global pheromone matrix, while slaves execute a savings-based ant colony algorithm Reimann et al. (2002) for the resulting restricted VRP.

Data decomposition methods induce different search behavior and solution quality compared to those of the sequential meta-heuristic. Such methods appear increasingly needed as the dimensions of contemplated problem instances continue to grow. Given the increased complexity of the problem settings, work is also required on how to best combine search-space decomposition and the other parallelization strategies, cooperation in particular. The Integrative Cooperative Search of Lahrichi et al. (2015) is a step in this directions (see Section 7 Advanced Cooperation Strategies - Creating Knowledge).

5 Independent Multi-search

Independent multi-search was among the first parallelization strategies proposed in the literature. It is also the most simple and straightforward p-control parallelization strategy and generally offers very interesting performances.

Independent multi-search seeks to accelerate the exploration of the search space toward a better solution (compared to sequential search) by initiating simultaneous solvers from different initial points (with or without different search strategies). It thus par-

allelizes the multi-start strategy by performing several searches simultaneously on the entire search space, starting from the same or from different initial solutions, and selecting at the end the best among the best solutions obtained by all searches. Independent multi-search methods thus belong to the pC/RS class of the taxonomy. No attempt is made to take advantage of the multiple solvers running in parallel other than to identify the best overall solution at the synchronization moment when all searches stop.

Independent multi-search turns out to be effective, simply because of the sheer quantity of computing power they allow one to apply to a given problem. Formal insights into the behavior of these strategies may be found in Battiti and Tecchiolli (1992), Taillard (1994), Stutzle (1998) and ten Eikelder et al. (1999). Empirical efficiency was shown by many contributions that took advantage of its simplicity of implementation and relatively good performance expectation. pC/RS/MPSS parallelizations of neighborhood-based meta-heuristics were thus proposed for, e.g., tabu search for the QAP (Battiti and Tecchiolli, 1992), VRP (Taillard, 1994; Rego and Roucairol, 1996; Talbi et al., 1998) and production planning (Bock and O., 2000); GRASP for the QAP (Li et al., 1994; Pardalos et al., 1992, 1995b), the Steiner problem (Martins et al., 2000, 1998), and the 2-path telecommunication network design (Ribeiro and Rosseti, 2002a,b,c); simulated annealing for graph partitioning (Banos et al., 2004b,a; Lee and Lee, 1996) and the TSP (Miki et al., 2003); and variable neighborhood search for the p-median problem García-López et al. (2002). Independent multi-search pC/RS/MPSS applications to non-genetic evolutionary methods have been proposed for scatter search (García-López et al., 2006, 2003), as well as for ant-colony optimization for set covering (Rahoual et al., 2002), the TSP (Stutzle, 1998) and the VRP (Doerner et al., 2006). Similar performance was observed for genetic methods with full-sized populations (Cohoon et al., 1991a,b), which avoided the premature convergence observed for pC/RS independent multi-search GA with small-sized populations obtained by separating the initial population among the independent GA searches (e.g., Herdy (1992); Schlierkamp-Voosen and Mühlenbein (1994)).

Independent multi-search offers an easy access to parallel meta-heuristic computation, offering a tool when looking for a "good" solution without investment in methodological development or actual coding. Independent multi-search methods are generally outperformed by cooperative strategies, however, the latter integrating mechanisms enabling the independent solvers to share, during their search, the information their exploration generates. As explained in the following sections, this sharing and the eventual creation of new information out of the shared one, yields in most cases a collective output of superior solutions compared to independent and sequential search.

6 Cooperative Search

Cooperative multi-search has emerged as one of the most successful meta-heuristic methodologies to address hard optimization problems (see, e.g., Talukdar et al. (2003); Alba (2005); Crainic (2005); Crainic and Hail (2005); Crainic (2008); Crainic and Toulouse (2008); Talbi (2006); Crainic and Toulouse (2010)). Cooperative search is based on harnessing the capabilities of several solution methods through *cooperation mechanisms* providing the means to share information while addressing the same problem instance (and create new information out of the exchanged data in advanced settings; see Section 7 Advanced Cooperation Strategies - Creating Knowledge).

Cooperative-search strategies are thus defined by the solver components engaged in cooperation, the nature of the information shared, and their interaction mechanism. The solvers define trajectories in the search space from possibly different initial points or populations, by using possibly different search strategies (including the same method with different parameter settings or populations). The information-sharing cooperation mechanism specifies how these independent solvers interact, how the exchanged information is used globally (if at all), and how each process acts on the received information, using it within its own search and, thus, transforming it before passing it to other solvers. As further detailed in the following sections, various cooperation mechanisms have been proposed: diffusion among "neighboring" solution methods arrayed in particular communication architectures, e.g., fine-grained, cellular GA (e.g., Cantú-Paz, 2005; Luque et al., 2005) and multi-level cooperative search (Toulouse et al., 1999b); direct exchanges among processes as in coarse-grain, island GA (Cantú-Paz, 2005; Luque et al., 2005), A-teams (Talukdar et al., 1998, 2003), and Collegial Asynchronous strategies (Crainic et al., 1997, 1996); indirect exchanges through a common data repository and management structure such as the *adaptive* (Rochat and Taillard, 1995; Badeau et al., 1997; Berger and Barkaoui, 2004) and *central memory* (Crainic et al., 1996, 1997, 2006a; Le Bouthillier et al., 2005; Jin et al., 2012, 2014) strategies. The global search behavior of the cooperative parallel meta-heuristic then emerges from the local interactions among the independent solvers, yielding a "new" meta-heuristic in its own right (Crainic and Toulouse, 2008). The similarity between this behavior and that of systems where decisions emerge from interactions among autonomous and equal "colleagues" has inspired the name *collegial* associated to cooperative-search strategies in the taxonomy used in this chapter.

The exchanged information has to be *meaningful* and *timely*. The goals are twofold. First, to enhance the performance of the receiving solvers. Second, to create a global image of the status of the cooperative search as "complete" as possible, which would provide the means to guide the global search toward better performance in terms of computing time and solution quality than the simple concatenation of the results of the individual solvers. Of course, one desires to achieve these goals without excessive overhead. Toulouse et al. (1996) proposed a list of fundamental issues to be addressed

when designing cooperative parallel strategies for meta-heuristics: What information is exchanged? Between what processes is it exchanged? When is information exchanged? How is it exchanged? How is the imported data used? Implicit in their taxonomy and explicitly stated in later papers, the issue of whether the information is modified during exchanges or whether new information is created completes this list.

"Good" solutions make up the most often exchanged type of information. This usually takes the form of the local current-best solution of a given solver or the overall best. The question of when to send such solutions has to be carefully addressed, however, particularly when the receiving process is supposed to act momentarily on the incoming information. One should thus avoid sending all local current-best solutions, particularly when the solver is performing a series of improving moves or generations, as solutions are generally "similar" and the receiving solvers have no chance to actually act on the incoming information. Sending the overall current-best solution to all cooperating solvers should also be avoided, as it rapidly decreases the diversity of the exploration and, thus, increases the amount of worthless computational work (many solvers will search within the same region) and brings an early "convergence" to a not-so-good solution. Sending out local optima only, exchanging groups of solutions, implementing randomized selection procedures (generally biased toward good or good-and-diverse solutions) of the solutions to share, and having the cooperating solvers treat differently the received information are among the strategies aimed at addressing these issues.

Context information may also profitably be shared, and integrated into the mechanisms used to guide the overall search. Context information is routinely collected by meta-heuristics during their search. It may consist in statistical information relative to the presence of particular solution elements in improving solutions (e.g., the medium and long-term memories of tabu search), the impact of particular moves on the search trajectory (e.g., the scores of the moves of large adaptive neighborhood search), population diversity measures, individual resilience across generations, etc. A limited number of studies indicate the interest of context-information exchanges (see Section 7 Advanced Cooperation Strategies - Creating Knowledge), but more research is needed on this topic.

Cooperating solvers may communicate and exchange information directly or indirectly. *Direct* exchanges of information occur either when the concerned solvers agree on a meeting point in time to share information, or when a solver broadcasts its information to one or several other solvers without prior mutual agreement. The latter case is generally not performing well, except when solvers include costly mechanisms to store such information without disturbing their own execution until ready to consider it.

Indirect exchanges of information are performed through independent data structures that become shared sources of information solvers may access according to their own internal logic to post and retrieve information. Such data structures are known under various names, e.g., *blackboard* in computer-science and artificial-intelligence vocabulary;

memory, *pool*, and *data warehouse* (the terms *reference* and *elite set* are also sometime used) in the parallel meta-heuristic literature . We use *memory* in the following.

Centralized memory is the implementation of choice reported in the literature. Distributed memory mechanisms may be contemplated, where a number of memories are inter-connected, each servicing a number of solvers. Such hierarchical structures, involving several layers of solvers and memories, appear interesting when a large number of processors is to be involved, for integrative cooperation strategies, or when computations are to take place on grids or loosely coupled distributed systems. Issues related to data availability, redundancy, and integrity must be then addressed, as well as questions relative to the balancing of workloads and the volume of information exchanged. More research in needed on this topic.

The logical intercommunication network corresponding to the selected cooperation strategy takes the form of a *communication graph*. A node of the graph represents a solver or a memory. Edges define the pairs of solvers or of a solver and a memory that may communicate directly. The projection of the communication graph on the physical interconnection topology of the parallel computer executing the parallel program - complete graph, ring, grid, torus, and star are most often encountered in the literature - is normally part of the implementation process.

When and how information is exchanged specifies how frequently cooperation activities are initiated, by whom, and whether all concerned solvers must simultaneously engage in communications or not. *Synchronous* and *asynchronous* communications are the two classes of communication exchanged, and are discussed in the following sections. The accumulated knowledge of the field indicates for both classes that exchanges should not be too frequent to avoid excessive communication overheads and premature "convergence" to local optima (Toulouse et al., 1996; Crainic et al., 1996, 1997).

We complete this section with three remarks. First, "simple" cooperation designs, based on synchronization or only exchanging current best solutions, for example, often appear biased toward intensifying the search in already-explored regions where interesting solutions have been identified. Diversification capabilities, e.g., probabilistic or diversity-driven selection of exchanged solutions, are thus an important component of cooperative p-control strategies.

One also observes that the main principles of cooperative parallel strategies are the same for neighborhood- and population-based meta-heuristics, even though denominations and implementation approaches may differ. One thus finds, for example, The terms *coarse* and *fine-grained island* are thus used to identify the amplitude of the population (large or small, down to single individual eventually, respectively) of participating solvers in genetic-based cooperation. Similarly, *multi colony* is the term generally used for cooperation in the ant-colony community. The next sections are thus structured around

classes of strategies rather than by meta-heuristic type.

Finally, one should notice that cooperation takes place at two different levels. The first is the *explicit* information sharing specified by the design of cooperation mechanism. *Implicit* cooperation makes up the second level, where information spreads across the cooperating solvers through a diffusion process and correlated interactions (Toulouse et al., 1999a, 2004, 1998, 2000). Implicit cooperation is **not** specified explicitly in the design of the algorithm. It is thus a bottom up, global emergent phenomenon produced by the correlated interactions among searches. Important research issues and challenges are related to how to harness indirect cooperation to enhance the optimization capabilities of cooperative search. For example, how should one select solvers and direct cooperation mechanisms to yield a system-wide emergent behavior providing an efficient exploration of the solution space from an optimization point of view? We believe learning and dynamic self-adaptation, at the level of both individual solvers and the cooperating meta-heuristic, to be part of the answer. Several other areas of research study systems displaying emergent behaviors, e.g., decentralized autonomic computing, social robotics, swarm intelligence, clustering logistics activities in supply chains, etc., and cross-fertilization appears promising. Empirical and theoretical research in this area should yield design principles and tools for more powerful (parallel) meta-heuristics.

6.1 Synchronous Cooperation

Synchronous cooperation follows a pC/KS scheme, with any of the SPDS, MPSS or MPDS search differentiation approaches, where the independent cooperating meta-heuristics synchronize at particular moments to initiate an information exchange phase. Synchronize here means that every solver but the last stops its activities and waits for all others to be ready. The synchronization moments may be generated based on conditions external to all solvers (e.g., number of iterations since the last synchronization) or detected by a specially-designated solver. The information exchange phase must be completed before any solver can restart its exploration from the respective synchronization point.

Synchronization may use a complete communication graph or a more restricted, less densely connected communication topology, e.g., a ring, torus, or grid graph. *Global* exchanges of information among all solvers take place in the former case, while information follows a *diffusion* process through direct *local* exchanges of information among neighboring processes. In all cases, the objective is to re-create a state of complete knowledge at particular moments in the global search and, thus, to hopefully guide it toward a coordinated evolution toward the desired solution to the problem. This goal is rarely attained, however, and the price in computing time efficiency may be significant, as communications cannot be initiated before the slowest solver is ready to start.

6.1.1 Global information exchanges

Many pC/KS cooperative search meta-heuristics in the literature implement the strategy according to the master-slave model. The master process, which may or may not include one of the participating solvers, initiates the other processes, stops them at synchronization points, gathers the information to be shared, updates the global data, decides on the termination of the search and, either effectively terminates it or distributes the shared information (a good solution, generally, the overall best solution in many cases) to the solvers for the continuation of the search.

The VNS pC/KS method for the p-median problem proposed in García-López et al. (2002) followed this idea, as well as the tabu search-based algorithms proposed for the TSP (Malek et al., 1989), the VRP (using ejection chains, Rego, 2001; Rego and Roucairol, 1996), the QAP (De Falco et al., 1994) and the task mapping problem (De Falco et al., 1995), the last two contributions attempting to overcome the limitations of the master-slave implementation by allowing processes, on terminating their local search phases, to synchronize and exchange best solutions with processes running on neighboring processors (this idea represents a step toward a "true" pC/KS design using a partial solution-diffusion process). This idea was also used to implement coarse-grained island-based cooperating genetic methods (Czech, 2000; Solar et al., 2002), where the master stopped the cooperating meta-heuristics to initiate a *migration* operator exchanging among the independent populations the best or a small subset of the best individuals in each. Applied to ant-colony systems (Drias and Ibri, 2003), this strategy divided the colony into several sub-colonies, each assigned to a different processor. Each independent ant-colony meta-heuristic sent to the master its best solution once its ants finished searching. The master updated the pheromone matrix and started a new search phase. A more sophisticated pC/KS approach was proposed in Niar and Fréville (1997) for the 0-1 Multi-dimensional Knapsack Problem, where the master dynamically adjusted the parameters of the cooperating tabu searches according to the results each had obtained so far. Computational results showed that this dynamic adjustment to be beneficial.

Alternatively, pC/KS schemes can be implemented in "true" collegial fashion by empowering each cooperating solver to initiate synchronization once it reaches a predetermined status. It then broadcasts its data, followed by similar broadcasts performed by the other solvers. Once all broadcasts are completed and information is shared, each solver performs its own import procedures on the received data and proceeds with its exploration of the search space until the next synchronization event.

Such an approach, exchanging the current best solution or group of solutions, was proposed for simulated annealing Diekmann et al. (1996), where the solvers transmitted their best solutions every n steps, and re-started the search after updating their respective best solutions (see also Lee and Lee (1992a,b, 1995, 1996) for the graph partitioning problem). For tabu search applied to location problems with balancing requirements (Crainic

et al., 1995, 1996), solvers synchronized after a number of iterations either pre-defined or dynamically determined. Most synchronous coarse-grained island genetic parallel methods applied this strategy, migration operators being applied at regular intervals, e.g., Wilkerson and Nemer-Preece (1998) for satisfiability problems (the best individual of each population migrated to replace the worst of the receiving population), Flores et al. (2003) for multi-objective telecommunication network design with migration following each generation, and Cohoon et al. (1987, 1991a,b); Lin et al. (1994); Hidalgo et al. (2003) for graph-partitioning, the later implementing a hierarchical method, where the fitness computation was performed at the second level (through a master-slave implementation; the overhead due to the parallelization of the fitness became significant for larger numbers of processors). A similar strategy was proposed for the multi ant-colony algorithms (Michels and Middendorf, 1999; Middendorf et al., 2002). Each colony has its own pheromone matrix and may (homogeneous) or may not (heterogeneous) use the same update rule. Colonies synchronize after a fixed number of iterations to exchange elite solutions that are used to update the pheromone matrix of the receiving colony.

Synchronization involved the exchange of not only good solutions but also of important search parameters in the pC/RS/MPDS parallel iterated tabu search proposed for the vehicle routing problem (VRP) (Cordeau and Maischberger, 2012). The iterated tabu solvers started from a different initial solution and used different search parameters, but synchronized the number of consecutive iterations without improvement used to determine the stopping moment of the individual improvement phases. This provided the means to more equally distribute the work among cooperating processes. The solvers exchanged their best solutions, each solver probabilistically selecting the working solution for the next improvement phase among the received ones and its own. This methods proved to be both flexible and efficient for several classes of routing problem settings with several depots, periodicity of demands, and time windows.

Most studies cite above compared several parallel strategy for the meta-heuristic and problem setting at hand (Cohoon et al., 1987, 1991a,b; Crainic et al., 1995, 1996; Lee and Lee, 1992a,b, 1995, 1996; Lin et al., 1994). They contributed to show that synchronous pC/KS strategies with global information exchanges outperform independent search approaches, as well as the sequential version, particularly with respect to solution quality. These studies also pointed out, however, the benefit of dynamically-determined synchronization points, as well as the superiority of asynchronous communications.

6.1.2 Diffusion

The previous strategies are based on global exchanges of information, gathered at synchronization points during the computation and distributed to all search threads. The interest of global information-sharing strategies resides in the best information available at the synchronization moment being available to all the solvers involved in cooperation.

The main drawback results from this same characteristic, however, as solvers relying heavily on the same information, set of best solutions in most cases, tend to focus on the same regions of the search space. This generally results in a search lacking in diversity that, more often than not, proves inefficient.

Synchronized cooperation strategies based on *diffusion* of information through local exchanges among "neighboring" solvers have therefore been proposed. Such approaches are defined on sparse communication graphs displaying a particular topology, e.g. such as ring, torus, or grid graphs, where each node is directly linked to only a few other nodes. A solver is then a neighbor of another one in this context if there is a direct link between the two nodes on which they run, that is, if they run are adjacent in the communication graph.

Synchronization still means that all solvers stop and exchange information, but here they perform it with their neighbors exclusively. Consequently, the quantity of information each solver processes and relies upon is significantly reduces, while the exchanges between non-adjacent solvers are performed at the speed of diffusion through possibly several chains of local exchanges and data modifications.

This idea has been much less explored as the global-exchange strategy, even though synchronous cooperative mechanisms based on local exchanges and diffusion have a less negative impact on the diversity of the search-space exploration. A number of applications were proposed with good results for coarse-grained (Calégari et al., 1997; Tongcheng and Chundi, 2002) and fine-grained (Alba and Dorronsoro, 2004; Dorronsoro et al., 2007; Folino et al., 1998b,a; Mühlenbein, 1989, 1991) genetic-based evolutionary methods, as well as for ant-colony optimization (Middendorf et al., 2002).

Cooperation based on asynchronous information sharing generally outperform synchronous methods, however, and are the topic of the next subsection.

6.2 Asynchronous Cooperation

Asynchronous strategies largely define the "state-of-the-art" in parallel multi-search metaheuristics. Solvers initiate asynchronous cooperation activities according to their own design logic and current internal state only, without coordination with other solvers or memories. Information sharing then proceeds either by direct inter-solver exchanges or through a data repository structure. These strategies belong to either the pC/C, this section, or the pC/KC, next section, collegial classes of the taxonomy, the latter using the shared information to generate new knowledge and global search guidance.

Two main benefits are obtained when relying on asynchronous communications. First, this provides the means to build cooperation and information sharing among solvers

without incurring the overheads associated to synchronization. Second, it increases the adaptability of cooperative meta-heuristics, as their capability to react and dynamically adapt to the exploration of the search space by the cooperating solvers is significantly increased compared to the other parallelization strategies. Of course, these benefits come with potential issues one must care for. For example, the information gathered during the search will seldom, if ever, be completely available when a process must decide. Also, too frequent data exchanges, combined to simple acceptance rules for incoming information, may induce an erratic solver behavior, the corresponding search trajectories becoming similar to random walks. Hence the interest for applying information-sharing quality, meaningfulness and parsimony principles (Crainic et al., 1996, 1997; Toulouse et al., 1996) .

In the basic asynchronous strategies discussed in this section, the shared information generally corresponds to a locally improving solution or individual(s). Most successful implementations have their cooperating solvers send out new local optima only. This limits the quantity of information sent and received, as well as the amount of work required to process it. Moreover, it avoids the case where a solver reorients it search from one of a series of improving solutions and ends up developing a trajectory similar to the one of the solver that originated the data.

The above-mentioned principles also explain the interest in diversifying the shared information (Crainic et al., 1996). Thus, always selecting the best available solution out of an elite set of good solutions, sent by potentially different solvers, proved less efficient in terms of quality of the final solution than a strategy that randomly, biased by quality, selected among the same elite set.

Finally, when to initiate cooperation activities and how to use incoming information is particular to each type of meta-heuristic involved in the cooperation. Yet, common to most strategies proposed in the literature is to perform jointly the sending and requesting of information. There is no absolute need to do this, however, even though it might decrease the amount of communication work. It might thus be interesting for neighborhood-based methods to make available right away their newly found local optima or improved overall solution, and not wait for the algorithmic step where examining external information is appropriate. Similarly, population-based methods could migrate a number of individuals when a significant improvement is observed in the quality and diversity of their elite group of individuals.

With respect to when to request external information, the parsimony principle implies limiting them to moments when the status of the search changes significantly, such as, when the best solution or the elite subpopulation did not improve for a number of iterations. The solver then engages into a so-called *search-diversification* phase, e.g., diversification in tabu search, change of neighborhood in variable neighborhood search, and complete or partial re-generation of population in population-based meta-heuristics,

involving the choice or modification of the solution to initiate the new phase. Examining the contribution of external information is appropriate in this context. Notice that it is always possible to use simply a pre-fixed number of iterations to initiating communications, but this approach should be restricted to meta-heuristics without search-diversification steps, e.g., tabu search based on continuous diversification.

Direct-exchange strategies are generally implemented over a complete communication graph, each solver sending out information to all other solvers or to a subset of them; this subset may be pre-defined or selected dynamically during the search. Particular communication graphs and information-diffusion processes could also be used but, despite encouraging results, too few experiments have been reported yet. We mention the work of Sevkli and Aydin (2007) proposing VNS pC/C strategies over uni and bidirectional ring topologies. Each solver was executing the basic VNS steps and on competing them, was passing its solution to its next neighbor (uni) or its two neighbors (bi), while receiving a solution fro its predecessor neighbor (uni) or its two neighbors (bi). The received solution was kept as initial solution of the next VNS run in the unidirectional case, while the best of the two received ones and the local one was kept in the bidirectional ring implementation. The latter strategy proved the most successful.

Information exchanges within pC/C strategies based on indirect communications are generally performed through a data repository structure, often called *central memory* (Crainic, 2005; Crainic et al., 1996, 1997). A solver involved in such a cooperation deposits (sends) good solutions, local optima generally, into the central memory, from where, when needed, it also retrieves information sent by the other cooperating solvers. The central memory accepts incoming solutions for as long as it is not full, acceptance becoming conditional to the relative interest of the incoming solution compared to the "worst" solution in the memory, otherwise. Evaluation is performed using the evaluation function for the global search space (or the objective function of the original problem). Diversity criteria are increasingly considered is this evaluation, a slightly worst solution being preferred if it increases the diversity of solutions in the central memory. Population culling may also be performed (deleting, e.g., the worst half the solutions in memory).

Both approaches may be applied to any meta-heuristic but, historically, most pC/C genetic-based evolutionary asynchronous cooperative meta-heuristics implemented a coarse-grained island model with with direct inter-solver exchanges. An early comparative study of coarse-grained parallel genetic methods for the graph-partitioning problem numerically showed the superiority of the pC/C strategy (with migration toward a subset of populations) over synchronous approaches (Lin et al., 1994).

The indirect-exchange communication model is found at the core of most asynchronous cooperative search strategies outside the genetic-evolutionary community, including simulated-annealing for graph partitioning (Lee and Lee, 1992a, 1995, 1992b, 1996) and the TSP (Sanvicente-Sánchez and Frausto-Solís, 2002), and VNS applied to

the VRPTW (Polacek et al., 2008) and the p-median problem (Crainic et al., 2004). A master process was associated to the central memory in the latter method, which kept, updated, and communicated the current overall best solution (it also initiated and terminated the algorithm). Individual solvers proceeded with the VNS exploration for as long as the solution was improved. When this was no longer the case, the current best was communicated to the master (if better than the one at the last communication) and the overall best solution was requested from it. The best solution between the local and imported ones was selected and the search was then continued in the current (local) neighborhood. Computational results on TSPLIB problem instances with up to 11849 customers showed that the cooperative strategy yielded significant gains in terms of computation time without loosing on solution quality.

Apparently, Crainic et al. (1996) proposed the first central-memory asynchronous tabu search. The tabu search solvers addressed a multi-commodity location problem with balancing requirements. Each sent to the memory its local-best solution when improved and imported a probabilistically-selected (rank-biased) solution from the memory before engaging in a diversification phase, This method outperformed in terms of solution quality the sequential version, several synchronous variants, and a broadcast-based asynchronous pC/C cooperative strategy. The same approach was applied to the fixed cost, capacitated, multicommodity network design problem with similar results (Crainic and Gendreau, 2002). Similar approaches were proposed for a broad range of problem settings, including the partitioning of integrated circuits for logical testing (Aiex et al., 1998), two-dimensional cutting (Blazewicz et al., 2004), the loading of containers (Bortfeldt et al., 2003), labor-constrained scheduling (Cavalcante et al., 2002), the VRPTW (Le Bouthillier and Crainic, 2005) and the capacitated VRP (Jin et al., 2012).

Solvers involved in pC/C strategies may not be restricted to a single meta-heuristic. Thus, the solvers in the two-phase approach of Gehring and Homberger (1997, 2001, 2002); Homberger and Gehring (1999) for the VRPTW first applies an evolution strategy to reduce the number of vehicles, followed by a tabu search to minimize the total distance traveled. A different version of the same idea may be found in Bastos and Ribeiro (1999) for the Steiner problem, where each phase of the two phase is designed as a pC/C asynchronous central memory strategy, only the change from one phase to the next being synchronized. Solvers run reactive tabu search and path relinking meta-heuristics in the first and second phases respectively.

The *multi-level cooperative search* proposes a different pC/C asynchronous cooperative strategy based on controlled diffusion of information Toulouse et al. (1999b). Solvers are arrayed in a linear, conceptually vertical, communication graph and a local memory is associated to each. Each solver works on the original problem but at a different level of aggregation (the solver on the conceptually first level works on the complete original problem) and communicates exclusively with the solvers directly above and bellow that is, at higher and lower aggregation levels respectively. The local memories are used

to make send information to the immediate neighbors and to access the incoming data from the same, at moments dynamically determined according to the internal logic of the respective solver. In the original implementation, solvers were exchanging improved solutions, incoming solutions not being transmitted further until modified locally for a number of iterations to enforce the controlled diffusion of information. Excellent results have been obtained for various problem settings including graph and hypergraph partitioning problems Ouyang et al. (2000, 2002), network design Crainic et al. (2006b), feature selection in biomedical data Oduntan et al. (2008), and covering design Dai et al. (2009). It is noteworthy that one can implement multi-level cooperative search using a central memory by adequately defining the communication protocols. Also not yet fully defined and tested, this idea is interesting as it opens the possibility of richer exchanges mechanisms combining controlled diffusion and general availability of global information.

The central-memory pC/C asynchronous cooperation strategy has proved worthy by several criteria. It yields high-quality solutions and is computationally efficient as no overhead is incurred for synchronization. It also helps to address the issue of "premature convergence" in cooperative search, by diversifying the information received by the participating solvers through probabilistic selection from the memory and by a somewhat large and diverse population of solutions in that central memory (solvers may thus import different solutions even when their cooperation activities are taking place within a short time span).

The performance of the central-memory cooperation and the availability of exchanged information (kept in the memory) has brought the question of whether one could design more advanced cooperation mechanisms taking advantage of the information exchanged among cooperating solvers. The pC/KC described in the next section are the result of this area of research.

7 Advanced Cooperation Strategies - Creating New Knowledge

Cooperation and, in particular, the memory-based asynchronous strategy was offering a rich framework to combine solvers of different meta-heuristic and exact types, together with a population of elite solutions of continuously increased quality. But, was it worthwhile the development effort?

An interesting proof-of-concept come from the study of Crainic and Gendreau (1999) combining a genetic-method solver and an asynchronous pC/C tabu search for multicommodity location-allocation with balancing requirements (Crainic et al., 1996). The tabu search solvers were aggressively exploring the search space, building the elite solution

population in the central memory. The genetic method initialized its population with the one in the central memory once it contained a certain number of solutions. Its aim was to create new solutions to hopefully enrich the quality and diversity of the solutions exchanged among the cooperating solvers. Asynchronous migration transferred the best solution of the genetic population to the central memory, as well as solutions from the central memory toward the genetic population. This strategy did perform well, especially on larger instances. Most importantly, it showed that, while the overall best solution was never found by the genetic solver, the GA-enhanced cooperation yielded higher-quality solutions compared to the initial cooperation. It appeared that the newly created solutions offered a more diverse set of diversification opportunities to the tabu search solvers, translating into a more diverse global search yielding better solutions.

The conclusion was not only that it is worthwhile to involve solvers of different types in the cooperation, but also that it is beneficial to create new solutions out of the set of elite solutions in the central memory. The new solutions are different from their parent solutions and are added to the central memory if they improve compared to them. The process is thus doubly beneficial as better solutions in the memory directly enhance the quality of the global search, while increasing the diversity of solutions in memory provides the opportunity for cooperating solvers to explore new regions of the search space.

A second idea on developing advanced cooperation mechanisms concerned the information that may be extracted from the exchanged solutions, and the context information, eventually. It has thus been observed that optimal or near-optimal solutions are often similar in the values taken by a large number of variables. Moreover, it is well-known in the meta-heuristic field that one can learn from the characteristics of the solutions generated during the search, out of the best ones in particular, and use this learning to guide the search (see, for example, the studies on memory and learning performed for tabu search (Glover and Laguna, 1997)). Applied to cooperative search, it appeared promising to apply these learning techniques to the elite solutions in the population gradually built in the central memory, and to use the resulting information to guide the search performed by the cooperating solvers.

Asynchronous cooperative strategies that include mechanisms to create new solutions and to extract information out of the exchanged solutions make up the p-control knowledge collegial (pC/KC) class. In most developments in this field, cooperating solvers work on the complete problem formulation and data. A recent research trend addresses rich multi-attribute problem settings and propose pC/KC strategies where different solvers work on particular parts of the initial problem or on integrating the resulting partial solutions into complete ones. The next subsections describe these two cases.

7.1 pC/KC with solvers working on the complete problem

Two main classes of pC/KC cooperative mechanisms are found in the literature differing in the information that is kept in memory. *Adaptive-memory* methods store partial elements of good solutions (Rochat and Taillard, 1995), while complete ones are kept in *central-memory* methods (Crainic, 2005; Crainic and Toulouse, 2003; Crainic et al., 1996). The latter method generalizes the former and, the vocabulary used in the various papers not withstanding, the two approaches are becoming increasingly unified.

The *adaptive-memory* terminology was coined by Rochat and Taillard (1995) (see also (Glover, 1996; Taillard et al., 1997, 1998)). The method was targeting the VRP and the VRPTW and it marked a changing point in the state-of-the-art at the time. The main idea is to keep in the memory the individual components (vehicle routes in the initial application) of the solutions produced by the cooperating solvers (tabu search methods in Rochat and Taillard (1995)). Two types of information were recorded for each solution element kept in memory, a frequency counter of its appearance in the best solutions encountered so far, and its rank within the elite population in memory based on the characteristics (mainly the objective function value) of the solution from which it as extracted. Solvers constructed new complete solutions out of randomly (rank biased) selected partial elements, improved these new solutions, and returned the best ones to the memory. The rank-biased random selection of elements assured that the new solution is composed in most cases of routes from different elite solutions, thus inducing a powerful diversification effect.

Several interesting developments were proposed and conclusions were drawn within the context of successful adaptive-memory pC/KC algorithms. A set-covering heuristic was thus proposed as selection mechanism for the elements (VRPTW routes) used by solvers to generate new initial solutions (Schulze and Fahle, 1999). This mechanism proved very successful and has been used several times since (e.g., Groër and Golden (2011)). A two-level parallel scheme was proposed for the real-time vehicle routing and dispatching (Gendreau et al., 1999). A pC/KC/MPSS cooperating adaptive-memory method made up the first level, while the slave processors attached to each solver, a tabu search method based on route decomposition (Taillard, 1993), made up the second level. The performance of this method is noteworthy also because while many papers mention the possibility of hierarchical parallel schemes, very few actually implementations are found in the literature. Also for the VRPTW, the adaptive-memory approach of Badeau et al. (1997) yielded a number of interesting findings relative to the implementation of cooperative methods. Thus, when individual solvers are fast, as is generally the case for routing problems, it is beneficial to run several solvers on the same processor and group the exchanges with the central-adaptive memory (avoiding or reducing access bottlenecks to the latter). On the other hand, running the memory and solvers on the same processor is to be avoided (the solver execution reduces the response efficiency of the memory).

Solvers in *central-memory* methods indirectly exchange complete elite solutions and context information through the central-memory data repository device. Solvers may include constructive, improving and post-optimization heuristics (e.g., Le Bouthillier and Crainic (2005); Le Bouthillier et al. (2005)), neighborhood (e.g., tabu search (Di Chiara, 2006; Jin et al., 2012, 2014)) and population-based methods (e.g., genetic algorithms (Le Bouthillier and Crainic, 2005; Le Bouthillier et al., 2005; Di Chiara, 2006) and path relinking (Crainic et al., 2006a)), as well as exact solution methods, on restricted versions of the problem, eventually. The particular solvers to include depend on the particular application. They should be efficient for the problem at hand. They should also contribute to built and enhance solutions that may contribute to enhance both the quality and the diversity of the elite population being built in the central memory.

The central memory keeps full solutions, solution attributes and context information, both received from the solvers and newly created out of the exchanged information. To more clearly distinguish between the data warehousing and the information creating functions of central-memory mechanisms, we define the *Search Coordinator* (*SC*) as the process in charge of the latter function. The simplest version of the SC was used in the pC/C strategies of the previous section, where solutions in memory were ordered (generally according to the value of the objective function) and rank-biased randomly extracted to answer solver requests. The functions of the SC in pC/KC methods include creating new solutions, extracting appropriate solution elements, building statistics on the presence and performance of solutions, solutions elements, and solvers (these belong to the family of memories well-known in the meta-heuristic community), creating the information to return when answering solver requests (the latter are the so-called *guidance mechanisms*).

The cooperative meta-heuristic proposed by Le Bouthillier and Crainic (2005) for the VRPTW used a simple pC/KC mechanisms. Four solvers, two simple genetic algorithms with order and edge recombination crossovers, respectively, and two tabu search methods that perform well sequentially, the Unified Tabu (Cordeau et al., 2001) and TABUROUTE (Gendreau et al., 1994). The solvers sent their improved best solutions to the central memory and requested solutions from the same when needed (the genetic algorithms for crossover operations, at regular intervals for the Unified Tabu and at diversification time for TABUROUTE). Besides ordering and selecting the solutions to return, the SC was only performing post-optimization (2-opt, 3-opt, or-opt, and ejection-chain procedures used to reduce the number of vehicles and the total traveled distance) on the received solutions. Without any calibration or tailoring, this algorithm proved to be competitive with the best meta-heuristics of its day in linear speedups.

A more complete SC was proposed in Le Bouthillier et al. (2005) also for the VRPTW. The goal was for a guidance mechanism that, first, extracted and returned to solvers meaningful information in terms of individual guidance and global search performance an, second, was independent of problem characteristics, routes in particular, and could

be broadly applied to network-based problem settings. To work toward the second goal, the SC mechanism targeted an atomic element on network optimization, the arc. The basic idea of the SC mechanism was that an arc that appears often in good solutions and less frequently in bad solutions may be worthy of consideration for inclusion in a tentative solution, and vice versa. To implement this idea, the authors considered "Appearance" was measured by means of frequencies of inclusion of arcs in the elite (e.g., the 10% best), average (between the 10% and 90% best), and worst (the last 10%) groups of solutions in the central memory. *Patterns* of arcs were then defined representing subsets of arcs (not necessarily adjacent) with similar frequencies of inclusion in particular population groups. Guidance was obtained by transmitting arc patterns to the individual solvers indicating whether the arcs in the pattern should be "fixed" or "prohibited" to intensify or diversify the search, respectively. The solvers accounted for the "fix" and "prohibit" instructions by using the patterns to bias the selection of arcs for move or reproduction operations A four-phase fixed schedule (two phases of diversification at the beginning to broaden the search, followed by two intensification phases to focus the search around promising regions) was used (see Le Bouthillier (2007) for a dynamic version of this mechanism). Excellent performances in terms of solution quality and computing efficiency were observed compared to the best-performing methods of the day.

A different SC was proposed in Jin et al. (2014) for the capacitated VRP with tabu search solvers. Solvers periodically (after a number of iterations or when the solution has not been improved for a number of iterations) sent best solutions and received a solution back from the central memory from which the search was resumed. The SC mechanism aimed to identify and extract information from the solutions in memory to guide solvers toward intensification and diversification phases. This was obtained by clustering solutions, dynamically when solutions were received, according to the number of edges in common. Thus solutions in a given cluster have a given number of edges in common and it is assumed to represent a region of the search space. Search history indicators were also associated to clusters giving the number of solutions in the cluster and the quality of the solutions. This information was used to infer how thoroughly the corresponding region had been explored and how promising it appeared. Clusters were actually sorted according to the average solution value of their feasible solutions. The cluster with the lowest average value, that is, with a large number of very good solutions, was selected for intensification, while the on with the lowest number of solutions was selected for diversification. A solution is selected in the corresponding cluster and it is sent to the requesting solver. Excellent results were obtained in terms of solution quality and computation effort (an almost linear speedup wa observed up to 240 processors) compared to the stat-o-the-art methods of the day (including the parallel method of Groër and Golden (2011)).

A pC/KC/MPDS method proposed in Groër and Golden (2011) for the VRP demonstrates how specialized solvers may address different issues in a cooperative meta-heuristic, including the generation of new knowledge. Two types of solvers were defined in this

scheme. The so-called heuristic solvers improved solutions received from the SC associated to the central memory (called master in Groër and Golden (2011)), through a record-to-record meta-heuristic (Chao et al., 1995; Golden et al., 1998; Li et al., 2005). On completing the task, solvers return both a number (50) of the best solutions found and the corresponding routes (a post-optimization procedure is run first o each route). Simultaneously, set-covering solvers aim to identify new solutions by solving series of set covering problems starting from a limited set of routes. Each time a set covering problem is solved, the solution is returned to the central memory and the set of the current 10 best solutions is retrieved and for the next run. Set-covering solvers have also access to the ordered list of best routes in memory and they select within to complete their problems. The number of routes admitted to set up a set covering problem is dynamically modified during the search to control the corresponding computational effort. The SC keeps and orders the received solutions and routes, and selects the solutions to make available to solvers (routes are always available; an efficient file system is used to facilitate access to this data). The method performed very well, both in terms of solution quality and computational effort (an almost-linear speedup was observed).

The contributions described in this section emphasize the interest of asynchronous knowledge-generating cooperative meta-heuristics. The cooperation and guidance mechanisms, as well as the role of learning an statistical performance data, require additional and systematic studies, preferably on a broader range of problem settings. The contributions aimed at addressing multi-attribute problem settings are described in the next subsection.

7.2 pC/KC with partial solvers - The Integrative Cooperative Search

The versatility and flexibility of the central-memory concept has raised the interest in generalizing it to address so-called *rich* combinatorial optimization problems displaying a large number of attributes characterizing their feasibility and optimality structures. The general idea is to decompose the initial problem formulation along sets of decision variables, called *decision-set attribute decomposition* in Lahrichi et al. (2015), yielding simpler but meaningful problem settings, in the sense that efficient solvers, can be "easily" obtained for these partial problems either by opportunistically using existing high-performing methods or by developing new ones. The central-memory cooperative search framework then brings together these partial problems and their associated solvers, together with integration mechanisms reconstructing complete solutions and search-guidance mechanisms.

The first effort in this direction is probably the work of Crainic et al. (2006a) (see also Di Chiara (2006)) for the design of wireless networks, where seven attributes were

considered simultaneously. The proposed pC/KC/MPDS cooperative meta-heuristic had tabu search solvers work on limited subsets of attributes only, while a genetic method amalgamated the partial solutions sent by the tabu search solvers to the central memory, into complete solutions to the initial problem.

The general method, called *Integrative Cooperative Search ICS*) by its authors has been fully defined in Lahrichi et al. (2015) (see also Crainic et al. (2009a,b)). We follow Lahrichi et al. (2015) in presenting an overview of ICS through an application to the multi-depot periodic vehicle routing problem (MDPVRP), which simultaneously decides on 1) selecting a visit *pattern* for each customer, specifying the particular periods the customer is to be visited over the multi-period planning horizon, and 2) assigning each customer to a depot for each visit (Mingozzi, 2005; Vidal et al., 2012).

The main components of ICS, which must be instantiated for each application, are the 1) decomposition rule; 2) *Partial Solver Groups* (*PSGs*) addressing the partial problems resulting from the decomposition; 3) *integrators*, which select partial solutions from PSGs, combine them to create complete ones, and sent them to the *Complete Solver Group* (*CSG*); and 4) the CSG, which corresponds to the central memory of ICS and has as prime function to manage the pool of complete solutions and the context information received from the PSGs and integrators, and to extract out of these the information required to guide the partial and global searches. Guidance is performed by the *Global Search Coordinator* (*GSC*) associated to the CSG. Notice that, in order to facilitate the cooperation, a unique solution representation is used throughout ICS. This representation is obtained by fixing rather than eliminating variables when defining partial problems.

The selection of the decision-sets is specific to each application case, decision variables being clustered to yield known or identifiable optimization problem settings. An opportunistic selection decomposes the MDPVRP along the depot and period decision sets to create two partial problems. Thus, ixing the customer-to-depot assignments yields a periodic VRP (PVRP), while fixing the patterns for all customers yields a multi-depot VRP (MDVRP). High-quality solvers exist in the literature for both problems.

Two PSGs are defined for the partial problems, one for the PVRP and the other for the MDVRP. Each PSG is organized according to the pC/KC paradigm and is thus composed of a set of *Partial Solvers*, a central memory where elite solutions are kept, and a *Local Search Coordinator* (*LSC*) managing the central memory and interfacing with the Global Search Coordinator.

Two algorithms were used in the implementation described in Lahrichi et al. (2015) for both complete or partial solvers, the HGSADC of Vidal et al. (2012) and GUTS, a generalized version of the Unified Tabu Search (Cordeau et al., 2001). Briefly, HGSADC combines the exploration capability of population-based evolutionary search, the aggressive-improvement strength of neighborhood-based local search to enhance solutions newly

created by genetic operators, and a solution evaluation function driven by both solution quality and contribution to the population diversity, which contributes to progress toward diverse and good solutions. GUTS is a tabu search-based meta-heuristic implementing advanced insertion neighborhoods and allowing the exploration of unfeasible solutions by dynamically adjusting penalties on violations of vehicle capacity and route duration constraints. Both methods use relaxation of vehicle-capacity and route-duration constraints combined to penalization of infeasibilities in the evaluation function. They also use well-known VRP local neighborhoods based on pattern-change, depot-change, inter and intra-route movements.

Integrators build complete solutions by mixing partial solutions with promising features obtained within the PSGs. Integrators aim for solution quality, the transmission of critical features extracted from the partial solutions, and computational efficiency. Several Integrators can be involved in an ICS implementation, contributing to these goals and increasing the diversity of the population of complete solutions.

The simplest Integrator consists in selecting high-quality partial solutions (with respect to solution value or the inclusion of particular decision combinations) and passing them directly to the Complete Solver Group. Meta-heuristics, population-based methods in particular, e.g., genetic algorithms (Vidal et al., 2012) and path relinking (Rahimi Vahed et al., 2013), my also be used, having proved their flexibility and stability in combining solution characteristics to yield high-quality solutions. Finally, a new methodology was proposed recently (El Hachemi et al., 2014). It proceeds through particular optimization models that preserve desired critical variables, defined as the variables whose values in the respective solution represent desired attributes, present in the partial solutions.

Four Integrators were included in the MDPVRP application, the simple one passing good solutions to the CSG, and three others starting from pairs of partial solutions randomly selected among the best 25% of the solutions in the central memories of the two PSGs. The second Integrator applied the crossover operator of HGSADC and enhanced the new solution through the local search education operator of the same method. The third and fourth Integrators applied the methodology of El Hachemi et al. (2014), the former aiming to transmit the attributes for which there is "consensus" in the input solutions, while the latter "promotes" them only through penalties added to the objective function.

The Complete Solver Group (CSG) includes a central memory, which includes the complete-solution set, as well as the context information and the guiding solutions built by the Global Search Coordinator (GSC). The CSG receives complete solutions from Integrators and, when solvers are included (e.g., GUTS and HGSADC in the present case), enhances them thus creating new ones. It is the Global Search Coordinator which builds the search contextual information (e.g., the frequency of appearance of each (customer, depot, pattern) triplet in the complete solution set, together with the cost of the

best solution containing it), 2) builds new guiding solutions to orient the search towards promising features, and 3) monitors the status of the solver groups, sending guiding instructions (solutions) when necessary.

Monitoring is performed by following the evolution of the PSGs by, e.g., interrogating the central memories of the PSGs for the number of improving solutions generated during a certain time period. Monitoring provides the means to detect undesired situations, e.g., loss of diversity in the partial or complete populations, stagnation in improving the quality of the current best solution, awareness that some zones of the solution space - defined by particular values for particular decision sets - have been scarcely explored, if at all, and that the search should be diversified in that direction, and so on. Whenever one of these criteria is not fulfilled, the GSC sends guidance "instructions" to the particular PSG. The particular type of guidance is application specific, but one may modify the values of the fixed attributes for the PSG to orient its search toward a different area or, more rarely, change the attribute subset under investigation (i.e., change the decomposition of the decision-set attributes), or modify/replace the solution method in a Partial Solver or Integrator.

In the present case, the GSC guided the search trajectory of a particular PSG by sending three solutions, which are either randomly selected (equiprobably) from the complete solution set, or are three *guiding* solutions built by the GSC. The receiving PSG adds directly these solutions to its own central memory, after resetting its population, all solutions being replaced by new randomly-generated ones. Guiding solutions were continuously generated, and stored in a particular pool, to reflect the current status and the history of the search represented by the context information. The process proceeded by selecting promising triplets with respect to the search history, that is, triplets that appear in at least one complete solution with a cost close (less than 3% distant) to the current best solution. The promising triplets are used to create feasible pattern and depot customer assignments, routes being then generated by the local search of HGSADC to complete the solutions. These solutions are then individually enhanced by a short execution of GUTS or HGSADC.

Extensive experimental analyses were conducted to 1) assess the performance of ICS when compared to state-of-the-art sequential methods and, 2) investigate a number of implementation alternatives.

The general conclusions were that ICS performed extremely well. It obtained very good results even when compared to the state-of-the-art HGSADC meta-heuristic, obtaining several new best-known solutions in shorter computing times. The experiments also indicated that 1) one should use solvers displaying similar time performances in order to have all solvers contributing reasonably equally to the cooperation; 2) when using genetic solvers in a PSG it is preferable for long runs to define a local population for each such solver and reserve the central memory of the PSG for communications and

guidance only, while using the central memory as population for all cooperating genetic solvers is better for short runs; and 3) embedding good solvers (HGSADC in the present case) in the CSG enhances slightly the already excellent performance of the ICS parallel meta-heuristic.

8 Perspectives

We presented a overview and state-of-the-art survey of the main parallel meta-heuristic ideas, discussing general concepts and algorithm design principle and strategies. The presentation was structured along the lines of a taxonomy of parallel meta-heuristics, which provides a rich framework for analyzing these design principles and strategies, reviewing the literature, and identifying trends and promising research directions.

Four main classes of parallel meta-heuristics strategies may be identified: low-level decomposition of computing-intensive tasks with no modification to the original algorithm, decomposition of the search space, independent multi-search, and cooperative (multi) search, the later encompassing synchronous, asynchronous collegial and knowledge-creating asynchronous collegial., It is noteworthy that this series also reflects the historical sequence of the development of parallel meta-heuristics. One should also note that, while the initial developments targeted genetic methods, simulated annealing, and tabu search, research is not addressing the full range of meta-heuristics. Furthermore, parallel meta-heuristics, cooperative search in particular, are know acknowledged as making up thir own class of meta-heuristics.

Many important research questions and challenges exist for parallel meta-heuristics, in terms of general design methodology, instantiation to particular meta-heuristic frameworks and problem settings, and implementation on various computing architectures.

It is indeed noteworthy that despite the many years of research on these issues, there are still many gaps in knowledge, as well as in the studied meta-heuristic frameworks and problem classes. One may single out the many variants of swarm-based optimization and non-genetic population based methods, scatter search and path relinking in particular. But one should not overlook the more classic meta-heuristic classes, as one still misses systematic and comprehensive / comparative studies of these issues. A large part of the studies present in the literature, targeted combinatorial optimization problems with relatively few attributes and a single level of decision variables, e.g., vehicle routing problems. This is to be understood, these problems being important for science and practice and displaying large search spaces. Significant less research has been dedicated to multi-attribute problem settings, like the rich VRP one increasingly has to tackle, and formulations with several levels of decisions like the single and multi-level network design.

We miss not only studies targeting particular meta-heuristic frameworks and problem classes, but also transversal studies comparing the behavior and performance of particular parallel meta-heuristic strategies over different problem classes, and of different parallel strategies and implementations for the same problem class. One increasingly finds such studies for sequential solution methods, we need them for parallel methods.

With respect to the four strategy classes, one should not forget that each fulfills a particular type of task and all are needed at some time. Thus, the idea that everything seems to be known regarding low-level parallelization strategies is no true. First, most studies on accelerating computing-intensive tasks targeted the evaluation of a population or neighborhood is classic meta-heuristic frameworks. These techniques should prove very valuable for swarm-based optimization and more research is required in this field. Second, as shown in recent studies, the best strategy to accelerate a local-search procedure may prove less effective when the local search is embedded into a full meta-heuristics or hierarchical solution methods. Third, the evolution of computing infrastructures opens up interesting but challenging perspectives. Let's emphasize the possibilities offered in particular by the Graphic Processing Units, which increase continuously in power and are present everywhere, as surveyed in Brodtkorb et al. (2013a,b).

Search-space decomposition also seems to have been thoroughly studied, and has been overlooked in the last years. Maybe due to the rapid and phenomenal increase in the memory available and the speed of access. Let's not forget, however, that most optimization problems of interest are complex and that the dimensions of the instances one faces in practice keep increasing. Research challenges exist in dynamic search-space decomposition and the combination of cooperative search and search-space decomposition. The Integrative Cooperative Search is a first answer in this direction, but more research is needed.

Asynchronous cooperation, particularly when relaying on memories as inter-solver communication mechanisms, provides a powerful, flexible and adaptable framework for parallel meta-heuristics that consistently achieved good results in terms of computing efficiency and solution quality for many meta-heuristic and problem classes. Other than the general research issues discussed above that are of particular interest in this context, a number of additional research issues and challenges are worth investigating.

A first issue concerns the exchange and utilization of context data locally generated by the cooperating solvers, to infer an image of the status of the global search and generate appropriate guiding instructions. Thus, contrasting the various local context data may be used to identify regions of the search space that were neglected or over explored. The information could also be used to evaluate the relative performance of the solvers conducting, eventually, to adjust the search parameters of particular solvers or even change the search strategy. So-called "strategic" decision variables or parameters could thus be more easily identified, which could prove very profitable in terms of search

guidance.

A related issue concerns the learning processes and the creation of new information out of the shared data. Important questions concern the identification of information that may be derived from the exchanged solutions and context information, and which is meaningful for, on the one hand, evaluating the status of the global search and, on the other hand, send to solvers to guide their own search as part of the global optimization effort. Research is this direction is still at the very beginning but has already proved its worth, in particular in the context of the integrative cooperative methods.

A third broad issue concerns the cooperation of different types of meta-heuristics and of these and exact solution methods. The so-called hybrid and matheuristic methods, representing the former and latter types of method combinations, respectively, are trendy in the sequential optimization field. Very few studies explicitly target parallel methods. How different methods behave when involved in cooperative search and how the latter behaves given various combinations of methods is an important issue that should yield valuable insights into the design of parallel meta-heuristic algorithms, Integrative Cooperative Search in particular. Actually, more research is required into ICS, both regarding its structure and components, and its application to various problem settings. A particularly challenging but fascinating direction for cooperative search and ICS is represented by the multi-scenario representation of stochastic optimization formulations, for which almost nothing beyond low-level scenario-decomposition has been proposed.

Finally, the issue of understanding cooperation on some fundamental level, giving the means to formally define and analyze itm in order to design better, more efficient algorithms. As mentioned earlier, this work parallels efforts in many other scientific domains addressing issues related to emerging decision and behavior out of the decisions and behaviors or several independent entities. Theoretical and empirical work is needed in order to address this fascinating and difficult question.

Acknowledgments

The author wishes to acknowledge the contributions of his colleagues and students, in particular Professors Michel Gendreau, Université de Montréal, Canada, and Michel Toulouse, the Vietnamese-German University, Vietnam, who collaborated over the years to the work on parallel meta-heuristics for combinatorial optimization. All errors are solely and entirely due to the author, however.

Partial funding for this project has been provided by the Natural Sciences and Engineering Council of Canada (NSERC), through its Discovery Grant and the Discovery Accelerator Supplements programs, and the Strategic Clusters program of the Fonds

québécois de la recherche sur la nature et les technologies. The author thanks the two institutions for supporting this research.

References

R.M. Aiex, S.L. Martins, C.C. Ribeiro, and N.R. Rodriguez. 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*, volume 1457 of *Lecture Notes in Computer Science*, pages 310–331. Springer-Verlag, 1998.

E. Alba, editor. *Parallel Metaheuristics: A New Class of Algorithms*. John Wiley & Sons, Hoboken, NJ, 2005.

E. Alba and B. Dorronsoro. Solving the Vehicle Routing Problem by Using Cellular Genetic Algorithms. In Gottlieb, J. and Günther, R.R., editors, *Evolutionary Computation in Combinatorial Optimization, 4th European Conference, EvoCOP 2004, Coimbra, Portugal, April 5-7, 2004*, volume 3004 of *Lecture Notes in Computer Science*, pages 11–20. Springer-Verlag, Heidelberg, 2004.

E. Alba, G. Luque, and S. Nesmachnow. Parallel Metaheuristics: Recent Advances and New Trends. *International Transactions in Operational Research*, 20(1):1–48, 2013.

R. Azencott. *Simulated Annealing Parallelization Techniques*. John Wiley & Sons, New York, NY, 1992.

P. Badeau, M. Gendreau, F. Guertin, J.-Y. Potvin, and E. Taillard. A Parallel Tabu Search Heuristic for the Vehicle Routing Problem with Time Windows. *Transportation Research Part C: Emerging Technologies*, 5(2):109–122, 1997.

R. Banos, C. Gil, J. Ortega, and F.G. Montoya. A Parallel Multilevel Metaheuristic for Graph Partitioning. *Journal of Heuristics*, 10(4):315–336, 2004a.

R. Banos, C. Gil, J. Ortega, and F.G. Montoya. Parallel Heuristic Search in Multilevel Graph Partitioning. In *Proceedings of the 12th Euromicro Conference on Parallel, Distributed and Network-Based Processing*, pages 88–95, 2004b.

R.S. Barr and B.L. Hickman. Reporting Computational Experiments with Parallel Algorithms: Issues, Measures, and Experts Opinions. *ORSA Journal on Computing*, 5 (1):2–18, 1993.

M.P. Bastos and C.C. Ribeiro. Reactive Tabu Search with Path-Relinking for the Steiner Problem in Graphs. In S. Voß, S. Martello, C. Roucairol, and Osman, I.H., editors, *Meta-Heuristics 98: Theory & Applications*, pages 31–36. Kluwer Academic Publishers, Norwell, MA, 1999.

R. Battiti and G. Tecchiolli. Parallel Based Search for Combinatorial Optimization: Genetic Algorithms and TABU. *Microprocessors and Microsystems*, 16(7):351–367, 1992.

J. Berger and M. Barkaoui. A Parallel Hybrid Genetic Algorithm for the Vehicle Routing Problem with Time Windows. *Computers & Operations Research*, 31(12):2037–2053, 2004.

J. Blazewicz, A. Moret-Salvador, and R. Walkowiak. Parallel Tabu Search Approaches for Two-Dimentional Cutting. *Parallel Processing Letters*, 14(1):23–32, 2004.

S. Bock and R. O. A New Parallel Breadth First Tabu Search Technique for Solving Production Planning Problems. *International Transactions in Operational Research*, 7(6):625–635, 2000.

A. Bortfeldt, H. Gehring, and D. Mack. A Parallel Tabu Search Algorithm for Solving the Container Loading Problem. *Parallel Computing*, 29:641–662, 2003.

A. R. Brodtkorb, T. R. Hagen, C. Schulz, and G. Hasle. GPU Computing in Discrete Optimization. Part I: Introduction to the GPU. *EURO Journal on Transportation and Logistics*, 2(1-2):129–157, 2013a.

A. R. Brodtkorb, T. R. Hagen, C. Schulz, and G. Hasle. GPU Computing in Discrete Optimization. Part II: Survey Focussed on Routing Problems. *EURO Journal on Transportation and Logistics*, 2(1-2):159–186, 2013b.

B. Bullnheimer, G. Kotsis, and C. Strauß. Parallelization Strategies for the Ant System. In R. De Leone, A. Murli, P. Pardalos, and G. Toraldo, editors, *High Performance Algorithms and Software in Nonlinear Optimization*, volume 24 of *Applied Optimization*, pages 87–100. Kluwer Academic Publishers, Dordrecht, 1999. URL `http://www.bwl.univie.ac.at/bwl/prod/papers/pom-wp-9-97.ps`.

P. Calégari, F. Guidec, P. Kuonen, and D. Kuonen. Parallel Island-Based Genetic Algorithm for Radio Network Design. *Journal of Parallel and Distributed Computing*, 47 (1):86–90, 1997.

E. Cantú-Paz. A Survey of Parallel Genetic Algorithms. *Calculateurs Parallèles, Réseaux et Systèmes répartis*, 10(2):141–170, 1998.

E. Cantú-Paz. Theory of Parallel Genetic Algorithms. In Alba, E., editor, *Parralel Metaheuristics: A New Class of Algorithms*, pages 425–445. John Wiley & Sons, Hoboken, 2005.

C.B.C. Cavalcante, V.F. Cavalcante, C.C. Ribeiro, and M.C. Souza. Parallel Cooperative Approaches for the Labor Constrained Scheduling Problem. In C. Ribeiro and P. Hansen, editors, *Essays and Surveys in Metaheuristics*, pages 201–225. Kluwer Academic Publishers, Norwell, MA, 2002.

J. Chakrapani and J. Skorin-Kapov. A Connectionist Approach to the Quadratic Assignment Problem. *Computers & Operations Research*, 19(3/4):287–295, 1992.

J. Chakrapani and J. Skorin-Kapov. Massively Parallel Tabu Search for the Quadratic Assignment Problem. *Annals of Operations Research*, 41:327–341, 1993a.

J. Chakrapani and J. Skorin-Kapov. Connection Machine Implementation of a Tabu Search Algorithm for the Traveling Salesman Problem. *Journal of Computing and Information Technology*, 1(1):29–36, 1993b.

I.M. Chao, B. L. Golden, and E.A. Wasil. An improved heuristic for the period vehicle routing problem. *Networks*, 26(1):25–44, 1995.

J. Cohoon, S. Hedge, W. Martin, and D. Richards. Punctuated Equilibria: A Parallel Genetic Algorithm. In J. Grefenstette, editor, *Proceedings of the Second International Conference on Genetic Algorithms and their Applications*, pages 148–154. Lawrence Erlbaum Associates, Hillsdale, NJ, 1987.

J. Cohoon, S. Hedge, and D. Richards. Genetic Algorithm and Punctuated Equilibria in VLSI. In Schwefel, H.-P. and Männer, R., editors, *Parallel Problem Solving from Nature*, volume 496 of *Lecture Notes in Computer Science*, pages 134–144. Springer-Verlag, Berlin, 1991a.

J. Cohoon, S. Hedge, and D. Richards. A Multi-Population Genetic Algorithm for Solving the k-Partition Problem on Hyper-Cubes. In R. Belew and L. Booker, editors, *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 134–144. Morgan Kaufmann, San Mateo, CA, 1991b.

J.-F. Cordeau and M. Maischberger. A parallel iterated tabu search heuristic for vehicle routing problems. *Computers & Operations Research*, 39(9):2033–2050, 2012.

J.-F. Cordeau, G. Laporte, and A. Mercier. A Unified Tabu Search Heuristic for Vehicle Routing Problems with Time Windows. *Journal of the Operational Research Society*, 52:928–936, 2001.

T.G. Crainic. Parallel Computation, Co-operation, Tabu Search. In C. Rego and B. Alidaee, editors, *Metaheuristic Optimization Via Memory and Evolution: Tabu Search and Scatter Search*, pages 283–302. Kluwer Academic Publishers, Norwell, MA, 2005.

T.G. Crainic. Parallel Solution Methods for Vehicle Routing Problems. In Golden, B.L., Raghavan, S., and Wasil, E.A., editors, *The Vehicle Routing Problem: Latest Advances and New Challenges*, pages 171–198. Springer, New York, 2008.

T.G. Crainic and M. Gendreau. Towards an Evolutionary Method - Cooperating Multi-Thread Parallel Tabu Search Hybrid. In S. Voß, S. Martello, C. Roucairol, and Osman, I.H., editors, *Meta-Heuristics 98: Theory & Applications*, pages 331–344. Kluwer Academic Publishers, Norwell, MA, 1999.

T.G. Crainic and M. Gendreau. Cooperative Parallel Tabu Search for Capacitated Network Design. *Journal of Heuristics*, 8(6):601–627, 2002.

T.G. Crainic and N. Hail. Parallel Meta-Heuristics Applications. In Alba, E., editor, *Parallel Metaheuristics: A New Class of Algorithms*, pages 447–494. John Wiley & Sons, Hoboken, NJ, 2005.

T.G. Crainic and M. Toulouse. Parallel Metaheuristics. In T.G. Crainic and G. Laporte, editors, *Fleet Management and Logistics*, pages 205–251. Kluwer Academic Publishers, Norwell, MA, 1998.

T.G. Crainic and M. Toulouse. Parallel Strategies for Meta-heuristics. In F. Glover and G. Kochenberger, editors, *Handbook in Metaheuristics*, pages 475–513. Kluwer Academic Publishers, Norwell, MA, 2003.

T.G. Crainic and M. Toulouse. Explicit and Emergent Cooperation Schemes for Search Algorithms. In Maniezzo, V., Battiti, R., and Watson, J.-P., editors, *Learning and Intelligent Optimization*, volume 5315 of *Lecture Notes in Computer Science*, pages 95–109. Springer-Verlag, Berlin, 2008.

T.G. Crainic and M. Toulouse. Parallel Meta-Heuristics. In Gendreau, M. and Potvin, J.-Y., editors, *Handbook of Metaheuristics (2nd Edition)*, pages 497–541. Springer, 2010.

T.G. Crainic, M. Toulouse, and M. Gendreau. Synchronous Tabu Search Parallelization Strategies for Multicommodity Location-Allocation with Balancing Requirements. *OR Spektrum*, 17(2/3):113–123, 1995.

T.G. Crainic, M. Toulouse, and M. Gendreau. Parallel Asynchronous Tabu Search for Multicommodity Location-Allocation with Balancing Requirements. *Annals of Operations Research*, 63:277–299, 1996.

T.G. Crainic, M. Toulouse, and M. Gendreau. Towards a Taxonomy of Parallel Tabu Search Algorithms. *INFORMS Journal on Computing*, 9(1):61–72, 1997.

T.G. Crainic, M. Gendreau, P. Hansen, and N. Mladenović. Cooperative Parallel Variable Neighborhood Search for the p-Median. *Journal of Heuristics*, 10(3):293–314, 2004.

T.G. Crainic, M. Gendreau, and J.-Y. Potvin. Parallel Tabu Search. In Alba, E., editor, *Parallel Metaheuristics*, pages 298–313. John Wiley & Sons, Hoboken, NJ, 2005.

T.G. Crainic, B. Di Chiara, M. Nonato, and L. Tarricone. Tackling Electrosmog in Completely Configured 3G Networks by Parallel Cooperative Meta-Heuristics. *IEEE Wireless Communications*, 13(6):34–41, 2006a.

T.G. Crainic, Y. Li, and M. Toulouse. A First Multilevel Cooperative Algorithm for the Capacitated Multicommodity Network Design. *Computers & Operations Research*, 33 (9):2602–2622, 2006b.

T.G. Crainic, G.C. Crisan, M. Gendreau, N. Lahrichi, and W. Rei. A Concurent Evolutionary Approach for Cooperative Rich Combinatorial Optimization. In *Genetic and Evolutionary Computation Conference - GECCO 2009, July 8-12, Montréal, Canada*. ACM, 2009a. CD-ROM.

T.G. Crainic, G.C. Crisan, M. Gendreau, N. Lahrichi, and W. Rei. Multi-thread Integrative Cooperative Optimization for Rich Combinatorial Problems. In *The 12th International Workshop on Nature Inspired Distributed Computing - NIDISC'09, 25-29 May, Rome*, 2009b. CD-ROM.

T.G. Crainic, T. Davidović, and D. Ramljak. Designing Parallel Meta-Heuristic Methods. In M. Despotovic-Zrakic, V. Milutinovic, and A. Belic, editors, *High Performance and Cloud Computing in Scientific Research and Education*, pages 260–280. IGI Global, 2014.

V.-D. Cung, S.L. Martins, C.C. Ribeiro, and C. Roucairol. Strategies for the Parallel Implementations of Metaheuristics. In C. Ribeiro and P. Hansen, editors, *Essays and Surveys in Metaheuristics*, pages 263–308. Kluwer Academic Publishers, Norwell, MA, 2002.

Z.J. Czech. A Parallel Genetic Algorithm for the Set Partitioning Problem. In *8th Euromicro Workshop on Parallel and Distributed Processing*, pages 343–350, 2000.

C. Dai, B. Li, and M. Toulouse. A Multilevel Cooperative Tabu Search Algorithm for the Covering Design Problem. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 68:35–65, 2009.

T. Davidović and T.G. Crainic. Parallel Local Search to Schedule Communicating Tasks on Identical Processors. *Parallel Computing*, 48:1–14, 2015.

I. De Falco, R. Del Balio, E. Tarantino, and R. Vaccaro. Improving Search by Incorporating Evolution Principles in Parallel Tabu Search. In *Proceedings International Confonference on Machine Learning*, pages 823–828, 1994.

I. De Falco, R. Del Balio, and E. Tarantino. Solving the Mapping Problem by Parallel Tabu Search. Report, Istituto per la Ricerca sui Sistemi Informatici Paralleli-CNR, 1995.

B. Di Chiara. *Optimum Planning of 3G Cellular Systems: Radio Propagation Models and Cooperative Parallel Meta-heuristics*. PhD thesis, Dipartimento di ingegneria dell'innovatione, Universitá degli Studi di Lecce, Lecce, Italy, 2006.

R. Diekmann, R. Lüling, B. Monien, and C. Spräner. Combining Helpful Sets and Parallel Simulated Annealing for the Graph-Partitioning Problem. *International Journal of Parallel Programming*, 8:61–84, 1996.

K. Doerner, R.F. Hartl, G. Kiechle, M. Lucka, and M. Reimann. Parallel Ant Systems for the Capacitated Vehicle Routing Problem. In Gottlieb, J. and Raidl, G.R., editors, *Evolutionary Computation in Combinatorial Optimization: 4th European Conference, EvoCOP 2004*, volume 3004 of *Lecture Notes in Computer Science*, pages 72–83. Springer-Verlag, Berlin, 2004.

K.F. Doerner, R.F. Hartl, and M. Lucka. A Parallel Version of the D-Ant Algorithm for the Vehicle Routing Problem. In Vajtersic, M., Trobec, R., Zinterhof, P., and Uhl, A., editors, *Parallel Numerics'05*, pages 109–118. Springer-Verlag, New York, NY, 2005.

K.F. Doerner, R.F. Hartl, S. Benkner, and M. Lucka. Cooperative Savings Based Ant Colony Optimization - Multiple Search and Decomposition Approaches. *Parallel Processing Letters*, 16(3):351–369, 2006.

M. Dorigo and T. Stuetzle. The Ant Colony Metaheuristic. Algorithms, Applications, and Advances. In F. Glover and G. Kochenberger, editors, *Handbook in Metaheuristics*, pages 251–285. Kluwer Academic Publishers, Norwell, MA, 2003.

B. Dorronsoro, F. Arias, A. Luna, A.J. Nebro, and E. Alba. A grid-based hybrid cellular genetic algorithm for very large scale instances of the CVRP. In W. Smari, editor, *High Performance Computing & Simulation Conference HPCS 2007 within the 21st European Conference on Modelling and Simulation ECMS 2007*, pages 759–765, 2007. http://www.scs-europe.net/conf/ecms2007/ecms2007-cd/ecms2007/ecms2007 final papers.html.

H. Drias and A. Ibri. Parallel ACS for Weighted MAX-SAT. In Mira, J. and Álvarez, J., editors, *Artificial Neural Nets Problem Solving Methods - Proceedings of the 7th International Work-Conference on Artificial and Natural Neural Networks*, volume 2686 of *Lecture Notes in Computer Science*, pages 414–421. Springer-Verlag, Heidelberg, 2003.

N. El Hachemi, T.G. Crainic, N. Lahrichi, W. Rei, and T. Vidal. Solution Integration in Combinatorial Optimization with Applications to Cooperative Search and Rich Vehicle Routing. Publication CIRRELT-2014-40, Centre interuniversitaire de recherche sur les réseaux d'entreprise, la logistique et le transport, Université de Montréal, Montréal, QC, Canada, 2014.

C.-N. Fiechter. A Parallel Tabu Search Algorithm for Large Travelling Salesman Problems. *Discrete Applied Mathematics*, 51(3):243–267, 1994.

C.D. Flores, B.B. Cegla, and D.B. Caceres. Telecommunication Network Design with Parallel Multi-objective Evolutionary Algorithms. In *IFIP/ACM Latin America Networking Conference 2003*, pages –, 2003.

G. Folino, C. Pizzuti, and G. Spezzano. Solving the Satisfiability Problem by a Parallel Cellular Genetic Algorithm. In *Proceedings of the 24th EUROMICRO Conference*, pages 715–722. IEEE Computer Society Press, 1998a.

G. Folino, C. Pizzuti, and G. Spezzano. Combining Cellular Genetic Algorithms and Local Search for Solving Satisfiability Problems. In *Proceedings of the Tenth IEEE International Conference on Tools with Artificial Intelligence*, pages 192–198. IEEE Computer Society Press, 1998b.

B.L. Garcia, J.-Y. Potvin, and J.M. Rousseau. A Parallel Implementation of the Tabu Search Heuristic for Vehicle Routing Problems with Time Window Constraints. *Computers & Operations Research*, 21(9):1025–1033, 1994.

F. García-López, B. Melián-Batista, J.A. Moreno-Pérez, and J.M.. Moreno-Vega. The Parallel Variable Neighborhood Search for the p-Median Problem. *Journal of Heuristics*, 8(3):375–388, 2002.

F. García-López, B. Melián-Batista, J.A. Moreno-Pérez, and J.M.. Moreno-Vega. Parallelization of the Scatter Search for the p-Median Problem. *Parallel Computing*, 29: 575–589, 2003.

F. García-López, M. García Torres, B. Melián-Batista, J.A. Moreno-Pérez, and J.M.. Moreno-Vega. Parallel Scatter Search. In Alba, E., editor, *Parallel Metaheuristics: A New Class of Metaheuristics*, pages 223–246. John Wiley & Sons, Hoboken, NJ, 2005.

F. García-López, M. García Torres, B. Melián-Batista, J.A. Moreno-Pérez, and J.M.. Moreno-Vega. Solving Feature Subset Selection Problem by a Parallel Scatter Search. *European Journal of Operational Research*, 169:477–489, 2006.

H. Gehring and J. Homberger. A Parallel Hybrid Evolutionary MetaHeuristic for the Vehicle Routing Problem with Time Windows. In K. Miettinen, M. M akel]:a, and J. Toivanen, editors, *Proceedings of EUROGEN99 - Short Course on Evolutionary Algorithms in Engineering and Computer Science*, pages 57–64. Jyvskyla, Finland, 1997.

H. Gehring and J. Homberger. A Parallel Two-Phase Metaheuristic for Routing Problems with Time Windows. *Asia-Pacific Journal of Operational Research*, 18(1):35–47, 2001.

H. Gehring and J. Homberger. Parallelization of a Two-Phase Metaheuristic for Routing Problems with Time Windows. *Journal of Heuristics*, 8:251–276, 2002.

M. Gendreau, A. Hertz, and G. Laporte. A Tabu Search Heuristic for the Vehicle Routing Problem. *Management Science*, 40:1276–1290, 1994.

M. Gendreau, F. Guertin, J.-Y. Potvin, and É.D. Taillard. Tabu Search for Real-Time Vehicle Routing and Dispatching. *Transportation Science*, 33(4):381–390, 1999.

M. Gendreau, G. Laporte, and F. Semet. A Dynamic Model and Parallel Tabu Search Heuristic for Real-time Ambulance Relocation. *Parallel Computing*, 27(12):1641–1653, 2001.

F. Glover. 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*, pages 1–75. Kluwer Academic Publishers, Norwell, MA, 1996.

F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, Norwell, MA, 1997.

B. L. Golden, E.A. Wasil, J.P. Kelly, and I.M. Chao. Metaheuristics in Vehicle Routing. In T. Crainic and G. Laporte, editors, *Fleet Management and Logistics*, pages 33–56. Kluwer Academic Publishers, Norwell, MA, 1998.

D.R.. Greening. A Taxonomy of Parallel Simulated Annealing Techniques. Technical Report No. RC 14884, IBM, 1989.

D.R. Greening. Parallel Simulated Annealing Techniques. *Physica D*, 42:293–306, 1990a.

D.R.. Greening. Asynchronous Parallel Simulated Annealing. *Lectures in Complex Systems*, 3:497–505, 1990b.

C. Groër and B. Golden. A parallel algorithm for the vehicle routing problem. *INFORMS Journal on Computing*, 23(2):315–330, 2011.

M. Herdy. Reproductive Isolation as Strategy Parameter in Hierarchical Organized Evolution Strategies. In R. Männer and B. Manderick, editors, *Parallel Problem Solving from Nature, 2*, pages 207–217. North-Holland, Amsterdam, 1992.

J.I. Hidalgo, M. Prieto, J. Lanchares, R. Baraglia, F. Tirado, and O. Garnica. Hybrid Parallelization of a Compact Genetic Algorithm. In *Proceedings of the 11th uromicro Conference on Parallel, Distributed and Network-Based Processing*, pages 449–455, 2003.

K. Holmqvist, A. Migdalas, and P.M. Pardalos. Parallelized Heuristics for Combinatorial Search. In A. Migdalas, P. Pardalos, and S. Storoy, editors, *Parallel Computing in Optimization*, pages 269–294. Kluwer Academic Publishers, Norwell, MA, 1997.

J. Homberger and H. Gehring. Two Evolutionary Metaheuristics for the Vehicle Routing Problem with Time Windows. *INFOR*, 37:297–318, 1999.

S. Janson, D. Merkle, and M. Middendorf. Parallel Ant Colony Algorithms. In Alba, E., editor, *Parallel Metaheuristics: A New Class of Metaheuristics*, pages 171–201. John Wiley & Sons, Hoboken, NJ, 2005.

J. Jin, T.G. Crainic, and A. Løkketangen. A Parallel Multi-Neighborhood Cooperative Tabu Search for Capacitated Vehicle Routing Problems. *European Journal of Operational Research*, 222(3):441–451, 2012.

J. Jin, T.G. Crainic, and A. Løkketangen. A Cooperative Parallel Metaheuristic for the Capacitated Vehicle Routing Problems. *Computers & Operations Research*, 44:33–41, 2014.

R. Laganière and A. Mitiche. Parallel Tabu Search for Robust Image Filtering. In *Proceedings of IEEE Workshop on Nonlinear Signal and Image Processing (NSIP'95)*, volume 2, pages 603–605, 1995.

N. Lahrichi, T.G. Crainic, M. Gendreau, W. Rei, C.C. Crisan, and T. Vidal. An Integrative Cooperative Search Framework for Multi-Decision-Attribute Combinatorial Optimization. *European Journal of Operational Research*, 2015. http://dx.doi.org/10.1016/j.ejor.2015.05.007.

P.S. Laursen. Parallel Heuristic Search – Introductions and a New Approach. In A. Ferreira and P. Pardalos, editors, *Solving Combinatorial Optimization Problems in Parallel*, volume 1054 of *Lecture Notes in Computer Science 1054*, pages 248–274. Springer-Verlag, Berlin, 1996.

A. Le Bouthillier. *Recherches coopératives pour la résolution de problèmes d'optimisation combinatoire*. PhD thesis, Département d'informatique et de recherche opérationnelle, Université de Montréal, Montréal, QC, Canada, 2007.

A. Le Bouthillier and T.G. Crainic. A Cooperative Parallel Meta-Heuristic for the Vehicle Routing Problem with Time Windows. *Computers & Operations Research*, 32(7):1685–1708, 2005.

A. Le Bouthillier, T.G. Crainic, and P. Kropf. A Guided Cooperative Search for the Vehicle Routing Problem with Time Windows. *IEEE Intelligent Systems*, 20(4):36–42, 2005.

K.G. Lee and S.Y. Lee. Efficient Parallelization of Simulated Annealing Using Multiple Markov Chains: An Application to Graph Partitioning. In Mudge, T.N., editor, *Proceedings of the International Conference on Parallel Processing*, volume III: Algorithms and Applications, pages 177–180. CRC Press, 1992a.

K.G. Lee and S.Y. Lee. Synchronous and Asynchronous Parallel Simulated Annealing with Multiple Markov Chains. In F. Brandenburg, editor, *Graph Drawing - Proceedings GD '95, Symposium on Graph Drawing, Passau, Germany*, volume 1027 of *Lecture Notes in Computer Science*, pages 396–408. Springer-Verlag, Berlin, 1995.

S.Y. Lee and K.G. Lee. Asynchronous Communication of Multiple Markov Chains in Parallel Simulated Annealing. In Mudge, T.N., editor, *Proceedings of the International Conference on Parallel Processing*, volume III: Algorithms and Applications, pages 169–176. CRC Press, Boca Raton, FL, 1992b.

S.Y. Lee and K.G. Lee. Synchronous and Asynchronous Parallel Simulated Annealing with Multiple Markov Chains. *IEEE Transactions on Parallel and Distributed Systems*, 7(10):993–1007, 1996.

F. Li, B. L. Golden, and E.A. Wasil. AVery large-scale vehicle routing: New test problems, algorithms, and results. *Computers & Operations Research*, 32(5):1165–1179, 2005.

Y. Li, P.M. Pardalos, and M.G.C. Resende. A Greedy Randomized Adaptive Search Procedure for Quadratic Assignment Problem. In *DIMACS Implementation Challenge, DIMACS Series on Discrete Mathematics and Theoretical Computer Science*, volume 16, pages 237–261. American Mathematical Society, 1994.

S.-C. Lin, W.F. Punch, and E.D. Goodman. Coarse-Grain Parallel Genetic Algorithms: Categorization and New Approach. In *Sixth IEEE Symposium on Parallel and Distributed Processing*, pages 28–37. IEEE Computer Society Press, 1994.

G. Luque, E. Alba, and B. Dorronsoro. Parallel Genetic Algorithms. In Alba, E., editor, *Parralel Metaheuristics: A New Class of Algorithms*, pages 107–125. John Wiley & Sons, Hoboken, 2005.

M. Malek, M. Guruswamy, M. Pandya, and H. Owens. Serial and Parallel Simulated Annealing and Tabu Search Algorithms for the Traveling Salesman Problem. *Annals of Operations Research*, 21:59–84, 1989.

S.L. Martins, C.C. Ribeiro, and M.C. Souza. A Parallel GRASP for the Steiner Problem in Graphs. In A. Ferreira and J. Rolim, editors, *Proceedings of IRREGULAR'98 – 5th International Symposium on Solving Irregularly Structured Problems in Parallel*, volume 1457 of *Lecture Notes in Computer Science*, pages 285–297. Springer-Verlag, 1998.

S.L. Martins, M.G.C. Resende, C.C. Ribeiro, and P.M. Parlados. A Parallel Grasp for the Steiner Tree Problem in Graphs Using a Hybrid Local Search Strategy. *Journal of Global Optimization*, 17:267–283, 2000.

R. Michels and M. Middendorf. An Ant System for the Shortest Common Supersequence Problem. In D. Corne, M. Dorigo, and F. Glover, editors, *New Ideas in Optimization*, pages 51–61. McGraw-Hill, 1999.

M. Middendorf, F. Reischle, and H. Schmeck. Multi Colony Ant Algorithms. *Journal of Heuristics*, 8(3):305–320, 2002. ISSN 1381-1231. doi: http://dx.doi.org/10.1023/A:1015057701750.

M. Miki, T. Hiroyasu, J. Wako, and T. Yoshida. Adaptive Temperature Schedule Determined by Genetic Algorithm for Parallel Simulated Annealing. In *CEC'03 - The 2003 Congress on Evolutionary Computation*, volume 1, pages 459–466, 2003.

A. Mingozzi. The multi-depot periodic vehicle routing problem. In *Abstraction, Reformulation and Approximation*, Lecture Notes in Computer Science, pages 347–350. Springer, Berlin / Heidelberg, 2005.

J.A. Moreno-Pérez, P. Hansen, and N. Mladenović. Parallel Variable Neighborhood Search. In Alba, E., editor, *Parallel Metaheuristics: A New Class of Metaheuristics*, pages 247–266. John Wiley & Sons, Hoboken, NJ, 2005.

H. Mühlenbein. Parallel Genetic Algorithms, Population Genetics and Combinatorial Optimization. In J. Schaffer, editor, *Proceedings of the Third International Conference on Genetic Algorithms*, pages 416–421. Morgan Kaufmann, San Mateo, CA, 1989.

H. Mühlenbein. Parallel Genetic Algorithms, Population Genetics, and Combinatorial Optimization. In Becker, J.D., I. Eisele, and Mündemann, F.W., editors, *Parallelism, Learning, Evolution. Workshop on Evolutionary Models and Strategies - WOPPLOT 89*, pages 398–406. Springer-Verlag, Berlin, 1991.

H. Mühlenbein. Parallel Genetic Algorithms in Combinatorial Optimization. In O. Balci, R. Sharda, and S. Zenios, editors, *Computer Science and Operations Research: New Developments in their Interface*, pages 441–456. Pergamon Press, New York, NY, 1992.

S. Niar and A. Fréville. A Parallel Tabu Search Algorithm For The 0-1 Multidimensional Knapsack Problem. In *11th International Parallel Processing Symposium (IPPS '97), Geneva, Switzerland*, pages 512–516. IEEE, 1997.

I. Oduntan, M. Toulouse, R. Baumgartner, C. Bowman, R. Somorjai, and T.G. Crainic. A Multilevel Tabu Search Algorithm for the Feature Selection Problem in Biomedical Data Sets. *Computers & Mathematics with Applications*, 55(5):1019–1033, 2008.

M. Ouyang, M. Toulouse, K. Thulasiraman, F. Glover, and J.S. Deogun. Multi-Level Cooperative Search: Application to the Netlist/Hypergraph Partitioning Problem. In *Proceedings of International Symposium on Physical Design*, pages 192–198. ACM Press, 2000.

M. Ouyang, M. Toulouse, K. Thulasiraman, F. Glover, and J.S. Deogun. Multilevel Cooperative Search for the Circuit/Hypergraph Partitioning Problem. *IEEE Transactions on Computer-Aided Design*, 21(6):685–693, 2002.

P.M. Pardalos, Y. Li, and M. K.A. Computational Experience with Parallel Algorithms for Solving the Quadratic Assignment Problem. In O. Balci, R. Sharda, and S. Zenios, editors, *Computer Science and Operations Research: New Developments in their Interface*, pages 267–278. Pergamon Press, New York, NY, 1992.

P.M. Pardalos, L. Pitsoulis, T. Mavridou, and M.G.C. Resende. 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*, volume 980, pages 317–331. Springer-Verlag, Berlin, 1995a.

P.M. Pardalos, Pitsoulis, L., and M.G.C. Resende. A Parallel GRASP Implementation for the Quadratic Assignment Problem. In A. Ferreira and J. Rolim, editors, *Solving*

Irregular Problems in Parallel: State of the Art, pages 115–130. Kluwer Academic Publishers, Norwell, MA, 1995b.

M. Polacek, S. Benkner, K.F. Doerner, and R.F. Hartl. A cooperative and adaptive variable neighborhood search for the multi depot vehicle routing problem with time windows. *Business Research*, 1(2):1–12, 2008.

S.C.S. Porto and C.C. Ribeiro. A Tabu Search Approach to Task Scheduling on Heteregenous Processors Under Precedence Constraints. *International Journal of High-Speed Computing*, 7:45–71, 1995.

S.C.S. Porto and C.C. Ribeiro. Parallel Tabu Search Message-Passing Synchronous Strategies for Task Scheduling Under Precedence Constraints. *Journal of Heuristics*, 1(2):207–223, 1996.

S.C.S. Porto, J.P.F..W. Kitajima, and C.C. Ribeiro. Performance Evaluation of a Parallel Tabu Search Task Scheduling Algorithm. *Parallel Computing*, 26:73–90, 2000.

A. Rahimi Vahed, T.G. Crainic, M. Gendreau, and W. Rei. A Path Relinking Algorithm for a Multi-Depot Periodic Vehicle Routing Problem. *Journal of Heuristics*, 19(3): 497–524, 2013.

M. Rahoual, R. Hadji, and V. Bachelet. Parallel Ant System for the Set Covering Problem. In M. Dorigo, G. Di Caro, and M. Sampels, editors, *Ant Algorithms - Proceedings of the Third International Workshop, ANTS 2002*, volume 2463 of *Lecture Notes in Computer Science*, pages 262–267. Springer-Verlag, Berlin, 2002.

D.J. Ram, T.H. Sreenivas, and K.G. Subramaniam. Parallel Simulated Annealing Algorithms. *Journal of Parallel and Distributed Computing*, 37:207–212, 1996.

M. Randall and A. Lewis. A Parallel Implementation of Ant Colony Optimisation. *Journal of Parallel and Distributed Computing*, 62:1421–1432, 2002.

C. Rego. Node ejection chains for the vehicle routing problem: Sequential and parallel algorithms. *Parallel Computing*, 27:201–222, 2001.

C. Rego and C. Roucairol. A Parallel Tabu Search Algorithm Using Ejection Chains for the VRP. In I. Osman and J. Kelly, editors, *Meta-Heuristics: Theory & Applications*, pages 253–295. Kluwer Academic Publishers, Norwell, MA, 1996.

M. Reimann, M. Stummer, and K. Doerner. A Savings Based Ants System for the Vehicle Routing Problem. In Langton, C., Cantú-Paz, E., Mathias, K.E., Roy, R., Davis, L., Poli, R., Balakrishnan, K., Honavar, V., Rudolph, G., Wegener, J., Bull, L., Potter, M.A., Schultz, A.C., Miller, J.F., Burke, E.K., and Jonoska. N., editors, *GECCO 2002: Proceedings of the Genetic and Evolutionary Computation Conference, New York, USA, July 9-13, 2002*, pages 1317–1326. Morgan Kaufmann Publishers, Inc., San Francisco, CA, 2002.

M. Reimann, K. Doerner, and R. Hartl. D-Ants: Savings Based Ants Divide and Conquer the Vehicle Routing Problem. *Computers & Operations Research*, 31(4):563–591, 2004.

C.C. Ribeiro and I. Rosseti. A Parallel GRASP Heuristic for the 2-path Network Design Problem. 4 journée ROADEF, Paris, February 20-22, 2002a.

C.C. Ribeiro and I. Rosseti. A Parallel GRASP Heuristic for the 2-path Network Design Problem. Third Meeting of the PAREO Euro Working Group, Guadeloupe (France), May, 2002b.

C.C. Ribeiro and I. Rosseti. Parallel grasp with path-relinking heuristic for the 2-path network design problem. AIRO'2002, L'Aquila, Italy, September, 2002c.

Y. Rochat and E. D. Taillard. Probabilistic Diversification and Intensification in Local Search for Vehicle Routing. *Journal of Heuristics*, 1(1):147–167, 1995.

H. Sanvicente-Sánchez and J. Frausto-Solís. MPSA: A Methodology to Parallelize Simulated Annealing and Its Application to the Traveling Salesman Problem. In C. Coello Coello, A. de Albornoz, L. Sucar, and O. Battistutti, editors, *MICAI 2002: Advances in Artificial Intelligence*, volume 2313 of *Lecture Notes in Computer Science*, pages 89–97. Springer-Verlag Heidelberg, 2002.

D. Schlierkamp-Voosen and H. Mühlenbein. Strategy Adaptation by Competing Subpopulations. In Y. Davidor, Schwefel, H.-P., and Männer, R., editors, *Parallel Problem Solving from Nature III*, volume 866 of *Lecture Notes in Computer Science*, pages 199–208. Springer-Verlag, Berlin, 1994.

J. Schulze and T. Fahle. A Parallel Algorithm for the Vehicle Routing Problem with Time Window Constraints. *Annals of Operations Research*, 86:585–607, 1999.

M. Sevkli and M.E. Aydin. Parallel variable neighbourhood search algorithms for job shop scheduling problems. *IMA Journal of Management Mathematics*, 18(2):117–133, 2007.

R. Shonkwiler. Parallel Genetic Algorithms. In S. Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 199–205. Morgan Kaufmann, San Mateo, CA, 1993.

M. Solar, V. Parada, and R. Urrutia. A Parallel Genetic Algorithm to Solve the Set-Covering Problem. *Computers & Operations Research*, 29(9):1221–1235, 2002.

T. Stutzle. Parallelization Strategies for Ant Colony Optimization. In Eiben, A.E., Back, T., Schoenauer, M., and Schwefel, H.-P., editors, *Proceedings of Parallel Problem Solving from Nature V*, volume 1498 of *Lecture Notes in Computer Science*, pages 722–731. Springer-Verlag, Heidelberg, 1998.

É.D. Taillard. Robust Taboo Search for the Quadratic Assignment Problem. *Parallel Computing*, 17:443–455, 1991.

É.D. Taillard. Parallel Iterative Search Methods for Vehicle Routing Problems. *Networks*, 23:661–673, 1993.

É.D. Taillard. Parallel Taboo Search Techniques for the Job Shop Scheduling Problem. *ORSA Journal on Computing*, 6(2):108–117, 1994.

É.D. Taillard, L.M. Gambardella, M. Gendreau, and J.-Y. Potvin. Adaptive Memory Programming: A Unified View of Metaheuristics. *European Journal of Operational Research*, 135:1–10, 1997.

É.D. Taillard, L.M. Gambardella, M. Gendreau, and J.-Y. Potvin. Programmation à mémoire adaptative. *Calculateurs Parallèles, Réseaux et Systèmes répartis*, 10:117–140, 1998.

E.-G. Talbi, editor. *Parallel Combinatorial Optimization*. Wiley-Interscience, Wiley & Sons, Hoboken, NJ, 2006.

E.-G. Talbi, Z. Hafidi, and J.-M. Geib. Parallel Adaptive Tabu Search Approach. *Parallel Computing*, 24:2003–2019, 1998.

E.-G. Talbi, O. Roux, C. Fonlupt, and D. Robillard. Parallel Ant Colonies for Combinatorial Optimization Problems. In J. R. et al., editor, *11th IPPS/SPDP'99 Workshops Held in Conjunction with the 13th International Parallel Processing Symposium and 10th Symposium on Parallel and Distributed Processing, April 12-16, San Juan, Puerto Rico*, volume 1586 of *Lecture Notes in Computer Science*, pages 239–247. Springer-Verlag, Berlin, 1999.

S. Talukdar, L. Baerentzen, A. Gove, and P. de Souza. Asynchronous Teams: Cooperation Schemes for Autonomous Agents. *Journal of Heuristics*, 4:295–321, 1998.

S. Talukdar, S. Murthy, and R. Akkiraju. Assynchronous Teams. In F. Glover and G. Kochenberger, editors, *Handbook in Metaheuristics*, pages 537–556. Kluwer Academic Publishers, Norwell, MA, 2003.

H.M.M. ten Eikelder, B.J.L. Aarts, M.G.A. Verhoeven, and E.H.L. Aarts. Sequential and Parallel Local Search for Job Shop Scheduling. In S. Voß, S. Martello, C. Roucairol, and Osman, I.H., editors, *Meta-Heuristics 98: Theory & Applications*, pages 359–371. Kluwer Academic Publishers, Norwell, MA, 1999.

G. Tongcheng and M. Chundi. Radio Network Design Using Coarse-Grained Parallel Genetic Algorithms with Different Neighbor Topology. In *Proceedings of the 4th World Congress on Intelligent Control and Automation*, volume 3, pages 1840–1843, 2002.

M. Toulouse, T.G. Crainic, and M. Gendreau. Communication Issues in Designing Cooperative Multi Thread Parallel Searches. In I.H. Osman and J.P. Kelly, editors, *Meta-Heuristics: Theory & Applications*, pages 501–522. Kluwer Academic Publishers, Norwell, MA, 1996.

M. Toulouse, T.G. Crainic, B. Sansó, and K. Thulasiraman. Self-Organization in Cooperative Search Algorithms. In *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics*, pages 2379–2385. Omnipress, Madisson, WI, 1998.

M. Toulouse, T.G. Crainic, and B. Sansó. An Experimental Study of Systemic Behavior of Cooperative Search Algorithms. In S. Voß, S. Martello, C. Roucairol, and Osman, I.H., editors, *Meta-Heuristics 98: Theory & Applications*, pages 373–392. Kluwer Academic Publishers, Norwell, MA, 1999a.

M. Toulouse, K. Thulasiraman, and F. Glover. Multi-Level Cooperative Search: A New Paradigm for Combinatorial Optimization and an Application to Graph Partitioning. In P. Amestoy, P. Berger, M. Daydé, I. Duff, V. Frayssé, L. Giraud, and D. Ruiz, editors, *5th International Euro-Par Parallel Processing Conference*, volume 1685 of *Lecture Notes in Computer Science*, pages 533–542. Springer-Verlag, Heidelberg, 1999b.

M. Toulouse, T.G. Crainic, and K. Thulasiraman. Global Optimization Properties of Parallel Cooperative Search Algorithms: A Simulation Study. *Parallel Computing*, 26 (1):91–112, 2000.

M. Toulouse, T.G. Crainic, and B. Sansó. Systemic Behavior of Cooperative Search Algorithms. *Parallel Computing*, 30(1):57–79, 2004.

M.G.A. Verhoeven and E.H.L. Aarts. Parallel Local Search. *Journal of Heuristics*, 1(1): 43–65, 1995.

T. Vidal, T.G. Crainic, M. Gendreau, N. Lahrichi, and W. Rei. A Hybrid Genetic Algorithm for Multi-Depot and Periodic Vehicle Routing Problems. *Operations Research*, 60(3):611–624, 2012.

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

R. Wilkerson and N. Nemer-Preece. Parallel Genetic Algorithm to Solve the Satisfiability Problem. In *Proceedings of the 1998 ACM symposium on Applied Computing*, pages 23–28. ACM Press, 1998.