1.	Record Nr.	UNINA9910437569503321
	Autore	Hamadi Youssef
	Titolo	Combinatorial search : from algorithms to systems / / Youssef Hamadi
	Pubbl/distr/stampa	Heidelberg [Germany] : , : Springer, , 2013
	ISBN	3-642-41482-6
	Edizione	[1st ed. 2013.]
	Descrizione fisica	1 online resource (xiii, 139 pages) : illustrations (some color)
	Collana	Gale eBooks
	Disciplina	004 004.0151 006.3 519.6
	Soggetti	Combinatorial optimization Computer algorithms Constraint programming (Computer science) Electronic information resource searching
	Lingua di pubblicazione	Inglese
	Formato	Materiale a stampa
	Livello bibliografico	Monografia
	Note generali	Description based upon print version of record.
	Nota di bibliografia	Includes bibliographical references.
	Nota di contenuto	Chap. 1 - Introduction -- Chap. 2 - Boosting Distributed Constraint Networks -- Chap. 3 - Parallel Tree Search for Satisfiability -- Chap. 4 - Parallel Local Search for Satisfiability -- Chap. 5 - Learning Variables Dependencies -- Chap. 6 - Continuous Search -- Chap. 7 - Autonomous Search -- Chap. 8 - Conclusion and Perspectives.
	Sommario/riassunto	Although they are believed to be unsolvable in general, tractability results suggest that some practical NP-hard problems can be efficiently solved. Combinatorial search algorithms are designed to efficiently explore the usually large solution space of these instances by reducing the search space to feasible regions and using heuristics to efficiently explore these regions. Various mathematical formalisms may be used to express and tackle combinatorial problems, among them the constraint satisfaction problem (CSP) and the propositional satisfiability problem (SAT). These algorithms, or constraint solvers, apply search space reduction through inference techniques, use activity-based heuristics to guide exploration, diversify the searches through frequent restarts, and often learn from their mistakes. In this book the author focuses on knowledge sharing in combinatorial search, the capacity to

generate and exploit meaningful information, such as redundant constraints, heuristic hints, and performance measures, during search, which can dramatically improve the performance of a constraint solver. Information can be shared between multiple constraint solvers simultaneously working on the same instance, or information can help achieve good performance while solving a large set of related instances. In the first case, information sharing has to be performed at the expense of the underlying search effort, since a solver has to stop its main effort to prepare and commu nicate the information to other solvers; on the other hand, not sharing information can incur a cost for the whole system, with solvers potentially exploring unfeasible spaces discovered by other solvers. In the second case, sharing performance measures can be done with little overhead, and the goal is to be able to tune a constraint solver in relation to the characteristics of a new instance – this corresponds to the selection of the most suitable algorithm for solving a given instance. The book is suitable for researchers, practitioners, and graduate students working in the areas of optimization, search, constraints, and computational complexity.

2. Record Nr.	UNINA9910158734703321
Autore	Sakr Sherif
Titolo	Large-scale graph processing using Apache Giraph / / by Sherif Sakr, Faisal Moeen Orakzai, Ibrahim Abdelaziz, Zuhair Khayyat
Pubbl/distr/stampa	Cham : , : Springer International Publishing : , : Imprint : Springer, , 2016
Edizione	[1st ed. 2016.]
Descrizione fisica	1 online resource (XXV, 197 p. 102 illus., 87 illus. in color.)
Disciplina	005.74
Soggetti	Database management
	Big data
	Data structures (Computer science)
	Database Management
	Big Data/Analytics
	Data Structures
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia

Nota di contenuto	1. Introduction -- 2. Getting started with Giraph -- 3. Giraph-In-Action: Implementing Popular Graph Algorithms using Giraph -- 4. Giraph Programming Optimizations: Tips and Tricks -- 5. Similar Systems to Giraph -- 6. Conclusions.
Sommario/riassunto	This book takes its reader on a journey through Apache Giraph, a popular distributed graph processing platform designed to bring the power of big data processing to graph data. Designed as a step-by-step self-study guide for everyone interested in large-scale graph processing, it describes the fundamental abstractions of the system, its programming models and various techniques for using the system to process graph data at scale, including the implementation of several popular and advanced graph analytics algorithms. The book is organized as follows: Chapter 1 starts by providing a general background of the big data phenomenon and a general introduction to the Apache Giraph system, its abstraction, programming model and design architecture. Next, chapter 2 focuses on Giraph as a platform and how to use it. Based on a sample job, even more advanced topics like monitoring the Giraph application lifecycle and different methods for monitoring Giraph jobs are explained. Chapter 3 then provides an introduction to Giraph programming, introduces the basic Giraph graph model and explains how to write Giraph programs. In turn, Chapter 4 discusses in detail the implementation of some popular graph algorithms including PageRank, connected components, shortest paths and triangle closing. Chapter 5 focuses on advanced Giraph programming, discussing common Giraph algorithmic optimizations, tunable Giraph configurations that determine the system's utilization of the underlying resources, and how to write a custom graph input and output format. Lastly, chapter 6 highlights two systems that have been introduced to tackle the challenge of large scale graph processing, GraphX and GraphLab, and explains the main commonalities and differences between these systems and Apache Giraph. This book serves as an essential reference guide for students, researchers and practitioners in the domain of large scale graph processing. It offers step-by-step guidance, with several code examples and the complete source code available in the related github repository. Students will find a comprehensive introduction to and hands-on practice with tackling large scale graph processing problems using the Apache Giraph system, while researchers will discover thorough coverage of the emerging and ongoing advancements in big graph processing systems.