A hybrid Lagrangean metaheuristic for single machine scheduling problem with sequence-dependent setup times and due dates

Thiago Henrique Nogueira[a,*], Gabriel Pinheiro Alves Santos[a], Carlos Roberto Venâncio de Carvalho[b], Martín Gómez Ravetti[b]

[a]*Departamento de Engenharia de Produção, Universidade Federal de Viçosa, Cx Postal 22, Cep 38810-000, Rio Paranaíba, Minas Gerais, Brazil*
[b]*Departamento de Engenharia de Produção, Universidade Federal de Minas Gerais, Av. Presidente Antônio Carlos, 6627, Cep 30161-010, Belo Horizonte, Minas Gerais, Brazil*

Abstract

In this article, a hybrid Lagrangean metaheuristic is proposed for single machine scheduling problems with sequence-dependent setup times and due dates. The objective function considered throughout this work, is to minimize the total tardiness. Related works and taxonomies for hybrid metaheuristics are analyzed, through a thorough historical overview. The proposed hybrid Lagrangean metaheuristic is a Lagrangean relaxation integrated with a metaheuristic. The algorithm uses the information of the Lagrangean multipliers to construct and perturb feasible solutions. The algorithm performance is compared with previous works and we find that the upper bounds obtained were particularly good, proving optimality for several instances and tight gaps for others. Furthermore, to the best of our knowledge, the proposed methodology presents generally better results than previous related works.

Keywords:
Hybrid Metaheuristic, Metaheuristic, Single Machine Scheduling Problem, Sequence-dependent Setup Times, Lagrangean Relaxation

---

*Corresponding author.
Email address: `thiagoh.nogueira@ufv.br` (Thiago Henrique Nogueira)

1. Introduction

Machine scheduling problems deal with the assigning of scarce resources in activities through time, aiming to minimize one or more objectives. This enormous family of problems is implicitly or explicitly present in many applications. From the operational research point of view, the first problems started to be identified and formulated back in the 1950's. This manuscript approaches one of their simplest scenarios, a single machine environment. However, when the presence of sequence-dependent setup times with due date is considered, the problem becomes a very hard combinatorial optimization problem. Therefore, the execution time is often drastically increased by the size of the instance. Several times, small instances or instances with moderate size cannot be solved in a reasonable computational time.

The hybrid metaheuristics arise as an option to achieve this purpose. The hybrid method consists of cooperative combination of methods, exact and/or approximated, and aims to absorb to the limit the potentialities of all the approaches (see *surveys* Alba (2005), Puchinger and Raidl (2005), Raidl (2006), Blum and Roli (2008), Raidl and Puchinger (2008), Jourdan et al. (2009), Raidl et al. (2010) and Blum et al. (2011)). Some of these hybrid metaheuristics, as explained by the authors, aim to provide optimal solutions in a reduced time; while others focus on obtaining better solutions than those achieved with pure metaheuristics. The structure and concepts of hybrid metaheuristics can be obtained with the surveys Talbi (2002), Dumitrescu and Stützle (2003), Danna and Le Pape (2004), Alba (2005), Cotta et al. (2005), Puchinger and Raidl (2005), Raidl (2006), Blum and Roli (2008), Ehrgott and Gandibleux (2008), Raidl and Puchinger (2008), Jourdan et al. (2009), Blum et al. (2010), Raidl et al. (2010) and Blum et al. (2011).

Talbi (2002) propose a new taxonomy for hybrid metaheuristics; while Cotta et al. (2005) and Jourdan et al. (2009) extend its work classifying applications and considering new integration approaches. Dumitrescu and Stützle (2003) describe combinations based on local search approaches, which are reinforced with the use of exact algorithms. Danna and Le Pape (2004) present a classification which resembles hybrid algorithms, including constraint programming as well. Alba (2005) analyzes related works with focus on the design of hybrid metaheuristics, defining concepts, performances, type of models, and applications. Puchinger and Raidl (2005) discuss the state of the art in the combination of exact and/or approximated algorithms to solve combinatorial optimization problems. Two distinct categories are defined: *Collaborative Combinations* - the algorithms exchange information, but they are not part of each other; and *Integrated Combinations* - an algorithm is incorporated and subordinated to another one.

Raidl (2006) reviews many well-known hybridization approaches which are classified based on several characteristics. The key components are considered a tool box to construct new effective hybrid metaheuristics. Raidl and Puchinger (2008) and Raidl et al. (2010) present a brief introduction of basic concepts about integer linear programming. They also examine existent techniques to combine exact and approximated methods. Blum and Roli (2008) firstly describe a general view of metaheuristics and discuss introductory concepts of hybrid metaheuristics and some applications. Ehrgott and Gandibleux (2008) present a survey with recent developments of hybrid metaheuristics to multi-objective optimization. Lastly, Blum et al. (2011) extend the proposition of Blum et al. (2010) discussing the principal concepts related to hybrid metaheuristics and provide a literature review considering some of the most important hybridization efforts.

The taxonomy proposed by Talbi (2002), and reinforced by Cotta et al. (2005) and Jourdan et al. (2009), is divided in "design issues" and "implementation issues". The first category concerns to the hybrid algorithm itself, which involves issues such as functionality and architecture of the algorithm. The authors define few categories: *Low-level* - a given function of the metaheuristic is replaced by another method; *High-level* - the different algorithms are self-contained; *Relay*- a set of methods is applied one after another; *Teamwork*- represents cooperative optimization models. From these categories, four cooperation classes are derived: *Low-level-Relay* (LRH), *Low-level-Teamwork* (LTH), *High-level-Relay* (HRH) and *High-level-Teamwork* (HTH).

Before analyzing the main related works addressing hybrid metaheuristics, regular metaheuristic methods will be named such as in Jourdan et al. (2009), and similarly to Puchinger and Raidl (2005): *Ant Colonies (AC)*, Colorni et al. (1991); *Descent Local Search (LS)*, Steiglitz and Papadimitriou (1982); *Genetic Algorithms (GA)*, Holland (1975); *Greedy Heuristic (GH)*, Lawler (1976); *Greedy Randomized Adaptive Search Procedure (GRASP)*, Feo and Resende (1995); *Neural Network (NN)*, White and Sofge (1992); *Iterated Local Search (ILS)*, Lourenço et al. (2003); *Simulated Annealing (SA)*, Kirkpatrick et al. (1983); *Tabu Search (TS)*, Glover (1989); and *Variable Neighborhood Search (VNS)*, Mladenović and Hansen (1997). Other methods will be defined as "Other Heuristic Methods" (OH), (see Osman and Laporte (1996) and Reeves (1993) for more information about approximated methods). Among the exact methods we highlight: *Branch-and-Bound (BB)*, *Dynamic Programming (DP)*, *Exact Resolution (ER) - MIP (Mixed Integer Programming), IP (Integer Programming), LP (Linear Programming)* and Exact Resolutions, *Lagrangean relaxation based methods (LR)*, and *Linear and*

Integer Programming Based Methods (LIPBM)), such as *Branch-and-Cut, Branch-and-Price,* and *Branch-and-Cut-and-Price*. Other methods will be defined as "Other Exact Methods (OE)", (see Nemhauser and Wolsey (1988) and Wolsey (2008) for more information about exact methods).

Table 1 aims at consolidating a panorama of the principal efforts in the field of hybrid metaheuristics over the last 10 years. For better visualization, the efforts will be highlighted only in the hierarchical level of design, which is composed by four classes (*Low-level-Relay* (LRH), *Low-level-Teamwork* (LTH), *High-level-Relay* (HRH) and *High-level-Teamwork* (HTH)), as proposed in Talbi (2002) and discussed in Cotta et al. (2005) and Jourdan et al. (2009). The table is composed by five columns. The first indicates the year of publication. The second, the characteristics (type) of the method used, where **PC** Pure Combination, means, only metaheuristics, and **MC** Mixed Combination, means, metaheuristics combined with exact methods. The third column describes the methods used with priority order, which is, $A + B$ the method A is executed before B or A has B as function. The fourth column has a hierarchical classification of the hybrid metaheuristics, the fifth column described the problem and the last column presents the referenced article.

Table 1 presents several works varying from theoretical works to practical applications in industrial and medical fields. Among the analyzed works, the hybrid metaheuristics classified as **HRH** and **LRH** are more common, about 80%. Furthermore, the mixed combination (**MC**) presents the majority of the efforts.

Among the works described in Table 1, the ones in the field scheduling are highlighted, namely: Danna et al. (2003) (Job Shop Scheduling Problem), Shi and Wang (2003) (Bi-objective Broadcast Scheduling Problem), Spina et al. (2003) (Single Line Scheduling Problem), Grosso et al. (2004) (Single-Machine Total Weighted Tardiness Scheduling Problem), Palpant et al. (2004) (Project Scheduling with Resource Constraints Problem), Angel and Bampis (2005) (Single-Machine Total Weighted Tardiness Scheduling Problem), Blum (2005a) (Open Shop Scheduling Problem), Beck (2007) (Job Shop Scheduling Problem), Dahal et al. (2007) (Scheduling in Power Systems Problem), Ravetti (2007) (Parallel Scheduling Problem), Chen et al. (2008) (Scheduling Problems), Meyer (2008) (Machine Scheduling Problem with Sequence-Dependent Setup Times Problem), Watson and Beck (2008) (Job Shop Scheduling Problem), de Paula et al. (2010) (Parallel Scheduling Problem), Hiermann (2012) (Multimodal Home-Health Care Scheduling Problem) Ravetti et al. (2012) (Parallel hybrid heuristics for the permutation flow shop problem) and Sioud et al. (2012) (Single Machine Scheduling Problem with Sequence-Dependent Setup Times). We reinforce that, the efforts of Grosso et al.

Table 1: Principal Articles about Hybrid Metaheuristics between 2003-2013

Year	Type	Method	Classification	Problem	Article
2003	MC	OE + OH	LRH	*Dynamic Plant Layout Problem*	Balakrishnan et al. (2003)
	MC	LR + TS	HRH	*Single Source Capacitate Location Problem*	Cortinhal and Captivo (2003)
	MC	GA + BB	LTH	*Generalized Schwefel Function and Rule Base Learning Problem*	Cotta and Troya (2003)
	MC	LIPBM + LS	LRH	*Job Shop Scheduling Problem*	Danna et al. (2003)
	PC	TS + OH	LRH	*Arc Routing Problem*	Greistorfer (2003)
	MC	LS + BB	LTH	*Flow shop problem*	Haouari and Ladhari (2003)
	MC	OE/(GA+OH)	LRH/LTH	*Traveling Salesman Problem*	Jahuira and Cuadros-Vargas (2003)
	MC	BB + OE	HRH	*Protein Structure Prediction Problem*	Klepeis et al. (2003)
	MC	ER + GRASP	HRL	*Linear Integer Programming Problem*	Neto and Pedroso (2003)
	MC	BB + NN	HRH	*Bi-objective Broadcast Scheduling Problem*	Shi and Wang (2003)
	MC	ER + GA	HRH	*Single Line Scheduling Problem*	Spina et al. (2003)
2004	MC	LIPBM + TS	HRH	*One-Dimensional Bin Packing Problem*	Alvim et al. (2004)
	PC	GRASP + OH	LTH	*Multiconstraint Knapsack Problem*	Barake et al. (2004)
	MC	(GA + MA) + BB	HRH	*Bi-objective M-machines Flow-Shop Problem*	Basseur et al. (2004)
	MC	BS + LS	LTH	*2-machine Flow Shop*	Della Croce et al. (2004)
				Incapacitated P-median Location Problem	
	MC	ER + GA	HRH	*Generalized Assignment Problem*	Feltl and Raidl (2004)
	MC	LS + OE	LRH/HTH	*MIP Problems*	Focacci et al. (2004)
	MC	ILS + OE	LRH	*Single-Machine Total Weighted Tardiness Scheduling Problem*	Grosso et al. (2004)
	PC	TS + LS	LRH	*Traveling Salesman Problem*	Kanazawa and Yasuda (2004)
	MC	ER + OH	HRH	*Prize-Collecting Steiner Tree Problem*	Klau et al. (2004)
	MC	GA + ER	LTH	*Gene Regulatory Network Problems*	Koduru et al. (2004)
	MC	OE/BB + GA	LTH/HRH	*MIP Problems*	Kostikas and Fragakis (2004)
	MC	GA + ER	HRH	*Markov Decision Processes Problem*	Lin et al. (2004)
	PC	LS + OH	LTH	*MIP Problems*	Lozano et al. (2004)
	MC	LS + OE	HTH	*Project Scheduling with Resource Constraints Problem*	Palpant et al. (2004)
	MC	OH + OE	LRH	*Car Sequencing Problem*	Perron et al. (2004)
	MC	LIPBM + GA	LTH	*2D Bin Packing Problem*	Puchinger and Raidl (2004)
2005	MC	ILS + OE	LRH	*Single-Machine Total Weighted Tardiness Scheduling Problem*	Angel and Bampis (2005)
	PC	AC + OH	LTH	*Open Shop Scheduling Problem*	Blum (2005a)
	PC	AC + OH	LRH	*MIP Problems*	Blum (2005b)
	MC	TS + ER	HRH	*Multiminima Continuous Functions Problem*	Chelouah and Siarry (2005)
	MC	LS + OE	LTH	*Routing Problem*	Cowling and Keuthen (2005)
	MC	OE + OH	HTH	*MIP Problems*	Danna et al. (2005)
	MC	OE + OH	LTH	*Dynamic Facility Layout Problem*	Dunker et al. (2005)
				with Unequal Sizes of Departments	
	MC	BB + OH	HTH	*Multidimensional Knapsack Problem*	Gallardo et al. (2005)
	MC	OE + OH	LRH	*Rectangle Packing Problem with General Spatial Costs*	Imahori et al. (2005)
	MC	OE + LIPBM	HTH	*Multidimensional Knapsack Problem*	Lichtenberger (2005)
	MC	SA + BB	HTH	*Zero-one LP Problems*	Nwana et al. (2005)
	MC	ER + TS	HRL	*MIP Problems*	Pedroso and Kubo (2005)
	MC	ER + OH	LRH	*Car Sequence Problem*	Prandtstetter and Raidl (2005)
	PC	OH +OH	LRH/HRH	*Global Optimization Problem*	Price et al. (2005)
	MC	LIPBM + OE	HTH	*Multidimensional Knapsack Problem*	Puchinger et al. (2005)
	MC	OE + OH	HTH	*Generalized Assignment Problem*	Rego (2005)
	MC	GA + OE	HRH	*MIP Problems*	Shi et al. (2005)
	MC	ER + TS	HRH	*Multidimensional 0-1 Knapsack Problem*	Vasquez and Vimont (2005)
2006	PC	TS + OH	HRH	*Directed Capacitated Vehicle Routing Problem*	De Franceschi et al. (2006)
	MC	OE + AC	HRH	*Project Portfolio Selection Problem*	Doerner et al. (2006)
	MC	TS + LR/OE	HTH	*MIP Problems*	Glover (2006)
	MC	SA + ER	LTH	*Irregular Strip Packing Problem*	Gomes and Oliveira (2006)
	MC	VNS + OH/OE	HRH	*MIP Problems*	Hansen et al. (2006)
	MC	LR + GA	HRH	*Prize Collecting Steiner Tree Problem*	Haouari and Chaouachi Siala (2006)
	MC	OE + LS	LRH	*Packing Problem*	Ibaraki and Nakamura (2006)
	MC	OH + VNS/ER	HTH	*Multidimensional 0-1 Knapsack Problem*	Puchinger et al. (2006)
	PC	TS + OH	HRH	*Pickup and Delivery Problem with Time Window*	Ropke and Pisinger (2006)
	MC	OE + OH	LRH	*MIP Problems*	Sniedovich and Voß (2006)
	PC	ILS + OH	LRH	*Quadratic Assignment Problem*	Stützle (2006)
	MC	ILS + ER	LRH	*One-dimensional cutting stock variant (pattern restricted problem)*	Umetani et al. (2006)
				(pattern restricted) Problem	
	MC	OE + OH	LRH	*MIP Problems*	Wilbaut et al. (2006)
2007	MC	AC + OE	HRH	*Job Shop Scheduling Problem*	Beck (2007)
	PC	OH + OH	HRH	*Shortest Common Supersequence Problem*	Blum et al. (2007)
	PC	OH + LS	HRH	*Capacitate P-Median Problem*	Chaves et al. (2007)
	MC	GA + ER	HTH	*Scheduling in Power Systems Problem*	Dahal et al. (2007)
	MC	OH + LIPBM	LRH	*Multidimensional 0-1 Knapsack Problem*	Gallardo et al. (2007)
				Shortest Common Supersequence Problem	
	MC	SS + ER	LTH	*Bicriteria 0, 1-Knapsack Problem*	Gomes da Silva et al. (2007)
	PC	VNS + VND	LRH	*Simple Plant Location Problem*	Hansen et al. (2007)
	MC	LR + OH	LRH	*Generalized Assignment Problem*	Jeet and Kutanoglu (2007)
	MC	OH + OE	LRH/HRH	*MIP Problems*	Kiziltan et al. (2007)
	PC	TS + OH	HRH	*Partitioning Problem*	Öncan et al. (2007)
	MC	LR + GA	LRH	*Knapsack Constrained Maximum Spanning Tree Problem*	Pirkwieser et al. (2007)
	MC	OE + OH	HTH	*Three-Stage Two-Dimensional Bin Packing Problem*	Puchinger and Raidl (2007)
	MC	LR + VNS/GRASP + VNS	LRH	*Parallel Scheduling Problem*	Ravetti (2007)
				with Sequence-Dependent Setup Times	
	MC	ER + AC	HRH	*Symmetric TSP Problem*	Reimann (2007)
	MC	OH + LIPBM	LRH	*Polishing Mixed Integer Programming Solutions*	Rothberg (2007)
	MC	OH + OE	LRH	*Graph Coloring Problem*	Trick and Yildiz (2007)
	MC	OE + OH	LRH	*Multiple-Drug Cancer Chemotherapy Schedule Problem*	Tse et al. (2007)

2008	PC	OH + AC	HRH	*Simple Assembly Line Balancing Problem*	Blum (2008)
	MC	OH + LIPBM	LRH	*Longest Common Subsequence Problem*	Blum et al. (2008)
				Shortest Common Supersequence Problem	
	MC	GA + OE	HRH	*Scheduling Problems*	Chen et al. (2008)
	MC	OH + OE	LRH	*Maximum Weight Set Packing Problem*	Eremeev (2008)
				Minimum Weight Set Partition Problem	
				Simple Plant Location Problem	
	MC	ER + OH	HRH	*MIP Problems*	Glover (2008)
	MC	ER + VNS	LRH	*Generalized Minimum Spanning Tree Problem*	Hu et al. (2008)
	MC	OE + OH	LRH	*Traveling Salesman Problem*	Hu and Raidl (2008)
	MC	OE + OH	LRH	*Multiple Sequence Alignment Problem*	Juang and Su (2008)
	MC	AC + OE	LRH	*Car Sequencing Problem*	Khichane et al. (2008)
	MC	LR + OH	HRH	*Real-World Fiber Optic Network Design Problem*	Leitner and Raidl (2008)
	PC	AC + OH	LRH	*Set Partitioning Problem*	Maniezzo and Roffilli (2008)
	MC	AC + OE	LRH	*Machine Scheduling Problem*	Meyer (2008)
				with Sequence-Dependent Setup Times	
	PC	OH + (VNS + VND)	HRH/LRH	*Finding Consensus Trees by Evolutionary*	Pirkwieser and Raidl (2008)
	PC	VNS + OH	LRH	*Car Sequencing Problem*	Prandtstetter and Raidl (2008)
	MC	VNS + VND	HRH	*MIP Problems*	Puchinger and Raidl (2008)
	MC	AC + OE	LRH	*Job Shop Scheduling Problem*	Watson and Beck (2008)
2009	MC	OE + OH	LRH	*Simple Assembly Line Balancing Problem*	Bautista and Pereira (2009)
	MC	OE + OH	LRH	*MIP Problems*	Birattari (2009)
	MC	OE + OH	LRH	*K-Cardinality Tree (KCT) Problem*	Blum and Blesa (2009)
	MC	LR + OH	LRH	*Single Source Capacitate Facility Location Problem*	Boschetti and Maniezzo (2009)
				Membership Overlay Problem	
	MC	OE + OH	LRH	*DNA Sequencing Problem*	Caserta and Voß (2009)
	PC	TS + OH	LRH	*Car Sequencing Problem and Graph Coloring Problem*	Hentenryck and Michel (2009)
	MC	LR + VNS	LRH	*Capacitated Connected Facility Location*	Leitner and Raidl (2009)
	MC	AC + OE	HRH	*Knapsack, Quadratic Assignment*	Lombardi et al. (2009)
				Maximum Independent Set Problems	
	MC	VNS + ER	HRH	*Periodic Vehicle Routing Problem with Time Window*	Pirkwieser and Raidl (2009a)
	MC	VNS + ER	HRH	*Periodic Vehicle Routing Problem with Time Window*	Pirkwieser and Raidl (2009b)
	MC	LIPBM + OH	LRH	*Warehouse Logistics*	Prandtstetter (2009)
				Reconstruction of Destroyed Paper Documents	
	PC	AC + VND/VNS + VND	LRH/LRH	*Reconstructing Cross Cut Shredded Text Documents*	Prandtstetter and Raidl (2009)
	MC	(VNS + VND) + DP	LRH	*Computing Tours in a Spare Parts Warehouse*	Prandtstetter et al. (2009)
	MC	OH/NN + OE	LRH/HRH	*Stochastic Problems*	Prestwich et al. (2009)
	MC	LIPBM + OH	LRH	*Multicommodity Flow Formulations*	Rei et al. (2009)
				of a Capacitate Network Design Problem	
	MC	OH + OH	LRH	*Stock (Re-) placements in Last-In,*	Ritzinger et al. (2009)
				First-Out Warehouses Problem	
	PC	VNS + OH	LRH	*Video-Server Load Re-Balancing Problem*	Walla et al. (2009)
	MC	ER + OH	HRH	*MIP Problems*	Wilbaut and Hanafi (2009)
2010	MC	OH + OE	LRH	*Multidimensional Knapsack Problem*	Al-Shihabi and Ólafsson (2010)
	MC	LR + OH	LRH	*Single Source Capacitate Facility Location Problem*	Boschetti et al. (2010)
	MC	OE + OH	LRH	*DNA Sequencing Problem*	Caserta and Voß (2010)
	MC	LIPBM + OH	LRH	*Bounded Diameter Minimum Spanning Tree Problem*	Gruber and Raidl (2010)
	MC	AC + OE	HRH	*MIP Problems*	Khichane et al. (2010)
	MC	LR + LS/OH	LRH	*Two Network Design Problems*	Leitner and Leitner (2010)
	PC	OH + AC	HRH	*Traveling Salesman Problem with Time Windows*	López-Ibáñez and Blum (2010)
	PC	ILS + OH	LRH	*MIP Problems*	Lozano and García-Martínez (2010)
	PC	OH + LS	-	*Continuous Optimization*	Molina et al. (2010)
	PC	TS + OH	LRH	*Tabu Search and Frequency Assignment Problem*	Montemanni and Smith (2010)
	MC	LR + VNS	LRH	*Parallel Scheduling Problems*	de Paula et al. (2010)
				with Sequence-Dependent Setup Times	
	PC	OH + OH	HRH	*Construct High-School Timetables*	Pimmer (2010)
	PC	TS + OH	HRH	*Periodic Location-Routing Problem*	Pirkwieser and Raidl (2010a)
	PC	VNS + OH	LRH	*Periodic Routing Problem*	Pirkwieser and Raidl (2010b)
	MC	VNS + ER/LIPBM + OE	HRH	*Periodic Vehicle Routing Problem with Time Windows*	Pirkwieser and Raidl (2010c)
	MC	LR + OH / OE + LS	LRH	*knapsack Constrained Maximum Spanning Tree Problem*	Puchinger et al. (2010a)
				Vehicle Routing Problem	
	MC	ER + OH	HRH	*Multidimensional 0-1 Knapsack Problem*	Puchinger et al. (2010b)
	PC	GRASP + OH	RHR	*Max-Min Diversity Problem*	Resende et al. (2010)
	MC	AC + OH	LRH	*Reconstructing Cross-Cut Shredded Text Documents Problem*	Schauer et al. (2010)
2011	MC	OH + ILS + OE	HRH	*Traveling Salesman Problem*	Applegate et al. (2011)
	PC	OH + OH	HRH	*Rooted Delay-Constrained Minimum Spanning Tree Problem*	Berlakovich et al. (2011)
	MC	OE + OH	LRH	*Blocks Relocation Problem*	Caserta et al. (2011)
	MC	ER + OH	HRH	*0-1 Multidimensional Knapsack Problem*	Hanafi and Wilbaut (2011)
	MC	OH + ER	LTH	*Generalized Minimum Spanning Tree Problem*	Hu and Raidl (2011)
	MC	SA + ER	HRH	*Train Scheduling Problem*	Jamili et al. (2011)
	MC	ER + GA	HRH	*Fuel Consumption Optimization of Hybrid Electric Vehicles*	Krenek (2011)
	PC	OH + OH	HRH	*Construct High-School Timetables*	Pimmer and Raidl (2011)
	PC	VNS + OH	LRH	*Vehicle Routing Problem with Compartments*	Pirkwieser et al. (2011a)
	PC	VNS + OH	LRH	*Vehicle Routing Problem with Compartments*	Pirkwieser et al. (2011b)
2012	PC	TS + OH	HRH	*Directed Profitable Rural Postman Problem*	Archetti et al. (2012)
	PC	TS + OH	LRH	*Capacitated Vehicle Routing Problem with Three-Dimensional Loading Constraints*	Bortfeldt (2012)
	MC	OH + ER	HRH	*Multimodal Home-Health Care Scheduling Problem*	Hiermann (2012)
	PC	SA + OE	HRH	*Train Scheduling Problem*	Jamili et al. (2012)
	MC	ER + GA	HRH	*Fuel Consumption Optimization of Hybrid Electric Vehicles*	Krenek et al. (2012)
	MC	OH + SA	HRH	*Optimal Power Flow Considering Prohibited Zones and Valve Point Effect*	Niknam et al. (2012)
	MC	ER +OH	HRH	*Bioinformatics and Transportation Problem*	Pirkwieser (2012)
	MC	OH + LS	HRH	*Parallel hybrid heuristics for the permutation flow shop problem*	Ravetti et al. (2012)
	PC	GA +OH	HRH	*Single Machine Scheduling Problem with Sequence-Dependent Setup Times*	Sioud et al. (2012)
	MC	ILS + VND + OE	LRH	*Heterogeneous Fleet Vehicle Routing Problem*	Subramanian et al. (2012)
2013	PC	GRASP + VNS	LRH	*Virtual Network Mapping Problem*	Inführ and Raidl (2013a)
	PC	GRASP + VNS	LRH	*Virtual Network Mapping Problem*	Inführ and Raidl (2013b)
	PC	VNS + OH	LRH	*Virtual Network Mapping Problem*	Inführ and Raidl (2013c)
	PC	SA + OH	HRH	*Heterogeneous Fleet Vehicle Routing Problems with Two-Dimensional Loading Constraints*	Leung et al. (2013)
	PC	OH + OH	LTH	*Multi-Objective Optimal Power Flow*	Narimani et al. (2013)
	MC	VNS + ER	LRH	*Train Scheduling Problem*	Raidl et al. (2013)
	PC	GRASP + VNS/OH	LRH	*Balancing Bicycle Sharing Systems Problem*	Rainer-Harbach et al. (2013a)
	PC	GRASP + VNS/OH	LRH	*Static Balancing of Bicycle Sharing Systems Problem*	Rainer-Harbach et al. (2013b)
	PC	OH + OH	LRH	*Three-Dimensional Loading Capacitated Vehicle Routing Problem*	Ruan et al. (2013)
	MC	OH + OE	HRH	*Integrated Timetable Based Design of Railway Infrastructure Problem*	Schöbel et al. (2013)
	PC	AC + OH	LRH	*MIP Problems*	Solnon (2013)
	MC	ILS + VND + OE	LRH	*Vehicle Routing Problem*	Subramanian et al. (2013)

(2004), Angel and Bampis (2005) and Sioud et al. (2012) are more similar to the problem study in this article.

Grosso et al. (2004) and Angel and Bampis (2005) use a method denominated "Iterated Dynasearch Neighborhood", which is a hybrid metaheuristic based on "Iterated Local Search". It is defined as a method that uses dynamic programming as an exploration strategy inside of "Iterated Local Search". Other approaches in scheduling use hybrid metaheuristics with ILS, they are Stützle (2006) on "Quadratic Assignment" Problem and Lozano and García-Martínez (2010) on "MIP Problems".

Ravetti (2007) and de Paula et al. (2010) present an important contribution with a variant of the Lagrangean Relaxation to obtain good limits to the problem of parallel machines scheduling with sequence-dependent setup time. In these efforts, the Lagrangean Relaxation is improved by the use of metaheuristics incorporated as an internal procedure of the Lagrangean Relaxation. This procedure is used to generate feasible solutions during the execution of the Lagrangean Relaxation. The authors discuss the importance of considering information from the Lagrangean Multipliers to obtain better feasible solutions.

Pirkwieser et al. (2007) present a similar work, using the Lagrangean Relaxation applied to the "Knapsack Constrained Maximum Spanning Tree problem", in which the Lagrangean Multipliers influence the procedures executed by the Genetic Algorithm. Other works use hybridization of metaheuristics, but with less integration of the Lagrangean Relaxation. These efforts are, namely, Cortinhal and Captivo (2003), (Single source capacitate location problem); Glover (2006), (MIP Problems); Haouari and Chaouachi Siala (2006), (Prize Collecting Steiner Tree Problem); Jeet and Kutanoglu (2007), (Generalized Assignment Problem); Leitner and Raidl (2008), (Real-World Fiber Optic Network Design Problem); Leitner and Raidl (2009), (Capacitated Connected Facility Location); Boschetti and Maniezzo (2009), (Single Source Capacitate Facility Location Problem and Membership Overlay problem); Boschetti et al. (2010), (Single Source Capacitate Facility Location Problem); Leitner and Leitner (2010), (Two Network Design Problems); and Puchinger et al. (2010a), (Knapsack Constrained Maximum Spanning Tree problem and Vehicle Routing Problem).

Jeet and Kutanoglu (2007), Boschetti and Maniezzo (2009) and Boschetti et al. (2010) propose a Lagrangean Metaheuristic. These authors emphasize the great potential of combining metaheuristics and Lagrangean Relaxation. This Lagrangean Metaheuristic consists in using metaheuristic techniques to obtain feasible solutions by Lagrangean Multipliers at each iteration of the method. The obtained solutions tend to be good limits. We remark that in these efforts the potential of information of the Lagrangean Multipliers is still not completely used. Part of our goals goes in this direction.

Blum et al. (2011) present a survey in which the authors make clear the potential of the Lagrangean Metaheuristic and emphasize the advantages of hybrid methods when compared to pure methods. Among these advantages, we remark: *(i)* it allows the improvement of lower and upper bounds together; *(ii)* it allows the evaluation of the quality of the obtained limits; *(iii)* it allows increasing the method with the addition of cuts. The authors claim that the Lagrangean Metaheuristics obtained better results when applied in MIP models where the solution of the Lagrangean Dual is rapidly obtained.

In this manuscript we propose a Hybrid Lagrangean Metaheuristic to the single machine scheduling problem with due dates and sequence-dependent setup times. The work has a similar line of thought than the Boschetti and Maniezzo (2009) and Boschetti et al. (2010). However, it emphasizes the use of Lagrangean Multipliers as discussed in Pirkwieser et al. (2007), Ravetti (2007) and de Paula et al. (2010). The proposed algorithm uses the information obtained from the Lagrangean multipliers to construct feasible solutions through the heuristic NEH and performs search procedure through a ILS framework. Instances from the literature are used to compare our results with other works.

2. Hybrid Metaheuristic Proposed

According to the taxonomic proposition of Talbi (2002) and Jourdan et al. (2009), the proposed algorithm can be defined as *LRH - Low-level-Relay*, as the Lagrangean Relaxation incorporates the constructive heuristic NEH (Nawaz Enscore Ham) and the metaheuristic ILS as internal procedures. NEH and ILS are executed sequentially under the Lagrangean Relaxation procedure. The focus of this proposed hybrid heuristic is on obtaining approximated solutions (Approximated Resolution), in which the cooperative methods act through all the search space (Global Cooperation) and solve the same optimization problem. Considering the classification proposed by Puchinger and Raidl (2005), the hybridization proposed in this study can be defined as an *Integrated Combination*, in which the *heuristic procedure is integrated to an exact algorithm*. Algorithm 1 describes the framework of the proposed methodology.

2.1. Mathematical Model

Nogueira et al. (2014) analyze different MIP (Mixed Integer Programming) formulations to the single machine scheduling problem with due date and sequence-

Algorithm 1 Hybrid Lagrangean Metaheuristic Framework

Initial Conditions

loop

Step 1: Gets the cumulative information of Lagrangean Multipliers by solving the Lagrangean Dual

Step 2: Creates a list of jobs in descending order of the information of Lagrangean Multipliers

Step 3: Applies the NEH heuristic to the list generating a refined feasible solution (x̃)

Step 4: Applies the ILS to the solution x̃, where its perturbation procedure considers the information of the Lagrangean Multipliers

end loop

Return x̃

dependent setup times, with the objective of minimizing the weighted completion time and the weighted tardiness. As expected, formulations based on "Time-Indexed" variables presented stronger limits. This study also demonstrates that "Time-Indexed" formulations can be represented by "Time-Indexed" variables, according to Sousa and Wolsey (1992), and "Arc-Time-Indexed" variables, according to Pessoa et al. (2010). Results indicate that the latter formulation is interesting for column-generation algorithms, due to the large number of $O(n^2h)$ columns (variables) and $O(nh)$ rows (constraints). On the other hand, the formulation of Sousa and Wolsey presents $O(nh)$ columns (variables) and $O(n^2h)$ rows (constraints), according to Nogueira et al. (2014), which is more interesting for rows-generation algorithms. Therefore, in this study we will work with the formulation of Sousa and Wolsey.

A set J of n jobs is considered, where each job must be processed exactly once in a single machine that can handle one job at time without preemption. For a given job j, let p_j be its processing time and d_j its due date. s_{ij} is also defined as the setup time needed to process the job j immediately after job i, and T_j the tardiness of job j. The planning horizon is discretized for each job i into the periods $0,\ldots,h_i$. The h_i constant has the value based in Nogueira et al. (2014) as:

$$h_i = \sum_{l \in J, l \neq i} s_l^{max} + \sum_{l \in J} p_l,$$

where $s_l^{max} = \max_{i \in J, i \neq l}\{s_{li}\}$. The set of periods is defined as $H = \{0,\ldots,\max_{i \in J}\{h_i\}\}$ and T_j is the job j tardiness in time period $t \in H$. The binary "Time-Indexed" variables, x_{jt}, are defined. x_{jt} is equal to 1 if job j starts at time t and equal to 0 otherwise. The constraint sets of the MIP formulation are defined as follows:

$$\text{Minimize} \sum_{j \in J} \sum_{t=0}^{h_j - p_j + 1} T_j x_{jt} \tag{1}$$

subject to

$$\sum_{t=0}^{h_j - p_j + 1} x_{jt} = 1 \;\; \forall \in J, \tag{2}$$

$$x_{jt} + \sum_{s=\max\{0, t-p_i-s_{ij}+1\}}^{\min\{t+p_j+s_{ji}-1, h_i-p_i+1\}} x_{is} \leq 1$$

$$\forall i, j \in J, i \neq j, \; t \in \{0, \ldots, h_j - p_j + 1\} \subset H, \tag{3}$$

$$\sum_{i \in J} \sum_{s=\max\{t-p_i-SMin_i+1, 0\}}^{\min\{t, h_i-p_i+1\}} x_{is} \leq 1 \;\; \forall t \in H, \tag{4}$$

$$T_j \geq \sum_{t=0}^{h_j - p_j + 1} (t + p_j - d_j) x_{jt} \;\; \forall j \in J, \tag{5}$$

$$T_j \geq 0 \;\; \forall j \in J, \tag{6}$$

$$x_{jt} \in \{0, 1\} \;\; \forall j \in J, \; t \in \{0, \ldots, h_j - p_j + 1\} \subset H. \tag{7}$$

The objective function (1) minimizes the total tardiness of each job on each time period. The constraint set (2) ensures that the processing of each job starts at only one time period in the machine. The constraint set (3) ensures that if the job j is scheduled in the time period t, no other job i ($i \neq j$) can be scheduled between $t - p_i - s_{ij} + 1$ and $t + p_j + s_{ji} - 1$ periods. The constraint set (4) ensures that when the integrality of variables x_{it} is relaxed, the number of assigned jobs $i \in J$ between $\max\{t - p_i - SMin_i + 1, 0\}$ and $\min\{t, h_i - p_i + 1\}$ is at most 1, where $SMin_i$ is the minimum setup time from $i \in J$ for any $j \in J, j \neq i$. If the variables x_{it} are not relaxed these proposed constraints are redundant. The constraint set (5) ensures that the tardiness of job j is greater than or equals to the difference between its start time plus its processing time and its due date. The constraint set

(6) ensures the positive domain of T_j. The constraint set (7) ensures the integrality domain of x_{jt}. Readers are referred to Nogueira et al. (2014), for a more detailed analysis on the model.

2.2. Lagrangean Relaxation

Briefly, the Lagrangean relaxation consists in turning a difficult problem into a manageable one by allowing the violation of a set of constraints at a certain price (Lagrangean multipliers), while maintaining the rest of the problem intact. The value of the objective function of this Lagrangean subproblem is a lower bound of the original problem (minimization problem). And through the use of heuristics it is also possible to obtain feasible solutions at each step of the solution algorithm. For better understanding, we refer to Fisher (1973), Geoffrion (1978),Minoux (1986), Lasdon (2002), Guignard (2003), Boschetti and Maniezzo (2009), Boschetti et al. (2010) and Blum et al. (2011).

Following the proposal of Boschetti and Maniezzo (2009), Boschetti et al. (2010) and Blum et al. (2011), we relax the scheduling constraints (3), and the Lagrangean subproblem (LP) can be rapidly solved. The constraints created to strengthen the model (3) are also dualized in order to improve the lower bound, reduce memory expenditures, and allow a simple method of resolution. The resulting model with the constraints 3 and 4 dualized is described below:

$$\begin{aligned}\text{(LP) Minimize} \sum_{j \in J} \sum_{t=0}^{h_j - p_j + 1} T_j x_{jt} \\ + \sum_{j \in J} \sum_{t=0}^{h_j - p_j + 1} \sum_{i \in J, i \neq j} \pi_{jit} (x_{jt} + \sum_{s=a}^{b} x_{is} - 1) \\ + \sum_{t \in H} \Lambda_t (\sum_{j \in J} \sum_{s=a'}^{b'} x_{js} - 1) \end{aligned} \tag{8}$$

subject to

$$\sum_{t=0}^{h_j - p_j + 1} x_{jt} = 1 \ \ \forall \in J, \tag{9}$$

$$T_j \geq \sum_{t=0}^{h_j - p_j + 1} (t + p_j - d_j) x_{jt} \ \ \forall j \in J, \tag{10}$$

$$\pi_{jit} \geq 0 \;\; \forall j, i \in J,\, j \neq i, t \in \{0,\ldots,h_j - p_j + 1\} \subset H, \tag{11}$$

$$\Lambda_t \geq 0 \quad t \in H, \tag{12}$$

$$T_j \geq 0 \;\; \forall j \in J, \tag{13}$$

$$x_{jt} \in \{0,1\} \;\; \forall j \in J,\, t \in \{0,\ldots,h_j - p_j + 1\} \subset H. \tag{14}$$

where,

$$a = \max\{t - p_i - s_{ij} + 1, 0\} \text{ and } b = \min\{t + p_j + s_{ji} - 1, h_j - p_j + 1\},$$

$$a' = \max\{t - p_j - SMin_j + 1, 0\} \text{ and } b' = \min\{t, h_j - p_j + 1\}.$$

The constraint set (3), with the Lagrangean multipliers π_{jit} and the constraint set (4), with the Lagrangean multipliers Λ_t, are added to the objective function (1), to obtain the objective function of the Lagrangean subproblem (8). Considering $x_{jt} = 1 \; \forall j \in J$ and $t \in H$ it is possible to define the cost β_{jt} of each job j at time t. These values are obtained by the expansion of the original objective function, considering $x_{jt} = 1$.

$$\beta_{jt} = \begin{cases} +T_j \\ +\sum_{i \in J, i \neq J} \pi_{jit} \\ +\sum_{i \in J, i \neq j} \sum_{s=a''}^{b''} (\pi_{ijs}) \\ +\sum_{s=t}^{\min\{t+p_j+SMin_j-1, h_j-p_j+1\}} \Lambda_s \end{cases}$$

where,

$$a'' = \max\{t - p_j - s_{ji} + 1, 0\},\, b'' = \min\{t + p_i + s_{ij} - 1, h_j - p_j + 1\}.$$

The β costs define the impact of allocating job j at time t. Besides containing the information about the original objective function, the β costs also posses the information about the Lagrangean multipliers. The constraint sets (10) and (13) ensure that only positive tardinesses are considered on the costs. Given that the tardiness of each job j (T_j) can be defined as $\max\{t+p_j-d_j,0\}$, these constraints may be discarded. Therefore, with the new cost β_{jt}, it is possible to solve the problem by simply deciding how to allocate each job in an instance of time t. The new objective function with the costs β_{jt}, defines the Lagrange subproblem (LP'), which can be rewritten as:

$$\text{(LP') Minimize} \sum_{j\in J} \sum_{t=0}^{h_j-p_j+1} (\beta_{jt}x_{jt} - \sum_{i\in J, i\neq j} \pi_{jit}) - \sum_{t\in H} \Lambda_t. \tag{15}$$

It is notable that the objective function (15) is decremented by two constants ($-\sum_{i\in J,i\neq j}\pi_{jit}$ and $-\sum_{t\in H}\Lambda_t$). Therefore, the previously described model is an assignment problem where each job j must be allocated at a time slot t. This problem, according to Ravetti (2007) and de Paula et al. (2010), can be solved by a simple inspection algorithm ($O(nh)$). In this manner, this formulation requires less memory than the original problem. This occurs due to the preprocessing of data and because not all of the variables and constraints need to be loaded.

2.3. Non-Delayed Relax-and-Cut

The covered Lagrangean Dual Problem has a huge number of dualized constraints. Therefore, it leads to a huge number of nonzero entries in the subgradient. It implies that the subgradient norm value becomes enormous in little iterations, resulting in a quicker reduction of the step value, jeopardizing the convergence of the method. The Non-Delayed-Relax-and-Cut (NDRC), arises as an alternative solution to this difficulty, which was presented by Lucena (1974) and applied in de Paula et al. (2010). The idea behind the NDRC is to dualize constraints "on-the-fly", that is, as they become violated, increasing the convergence by decreasing the number of dualized constraints. In this manner, the method proposes the dualization of violated constraints after each iteration of the resolution algorithm of the Lagrangean dual problem (in our case the Volume Algorithm).

In the case of the proposed Lagrangean relaxation, the constraint sets (3) and (4) are relaxed. There is a small number of constraints (4) (h) compared to constraints (3) (hn^2). Thus, the structure of NDRC is applied only to the constraint set (3). Considering this constraint set, there are three possibilities: when no job is being processed in a given time interval, the subgradient will be negative (the constraints are not violated, but contributes to calculate the size of the step - equation 19); when only one job is being processed in a given time period, the subgradient will be null (the constraints are not violated); when the number of jobs processed is higher than 1 (one), in a given time interval, the subgradient will be positive (the constraints are violated).

The NDRC defines that in each iteration k of the Volume Algorithm (Section 2.4), the constraints may be found in three different groups: *Currently Violated Active Set*, (CA^k); *Previously Violated Active Set*, (PA^k); and *Currently Inactive Set*, (CI^k). The latter did not contribute to the value of the objective function of the Lagrangean relaxation problem. Nevertheless, they will interfere when determining the step of the Volume Algorithm, (see equation (19)). Besides that fact, when Lagrangean multipliers related to these constraints are null in the previous iteration, they continue null in the current iteration, as noted in the formula described in the Volume Algorithm ($\pi_k = \pi^{'} + sg_k$). Thus, as we are expecting to work with a large number of relaxed constraints in CI^k, the subgradient will be set to zero whenever the Lagrangean multiplier (π_k) is null, as described by Lucena (1974).

However, a large number of constraints of type $CA^k \backslash PA^k$ are expected, (violated in iteration k), that is, currently violated constraints with null Lagrangean multipliers associated to them. These constraints will become effectively dualized at the end of the k^{th} Volume algorithm iteration. In order to avoid the dualization of a large number of these constraints (reducing the convergence of the method), the algorithm Non-Delayed-Relax-and-Cut proposed dualizes just one new constraint ($CA^k \backslash PA^k$) for each job $j \in J$ at each iteration k. The constraint dualized will be the one to present the higher violation for a given job. Other constraints will be considered as inactive (CI^k), that is, their subgradient entries arbitrarily be set to zero. This procedure aims to improve the convergence of the proposed algorithm (see (de Paula et al., 2007, 2010)).

2.4. Volume Algorithm

The Lagrangean Dual is solved by the Volume algorithm such as proposed in Barahona and Anbil (2000). Its efficiency is demonstrated in Barahona and Ladányi (2006) and Fukuda (2007). In this work, the procedure is adapted by the Non-Delayed-Relax-and-Cut method, in a similar fashion to de Paula et al. (2010). The Volume algorithm is an extension to the subgradient algorithm, which will produce an approximation to primal and dual solutions. This algorithm has similar computational effort with the subgradient algorithm and it presents similarities with the Conjugate Subgradient method (Lemaréchal, 1975, Wolfe, 1975) and the Bundle method (Lemaréchal, 1989, Lemaréchal, 2001). The global convergence of a specific implementation of Volume algorithm is analyzed in Bahiense et al. (2002).

As in the proposed algorithm the precedence constraints are dualized, solutions tend to distribute the jobs in the first time slot of the machine, on the contrary the Lagrangean multipliers (β) tend to separate pairs of jobs. As the Lagrangean multipliers define a penalty to a given job allocated at a given position, if the job has a large associated value, it means that it has greater impact on the objective function, i.e., it is not allocated in a favorable position. And that information can be used to decide when to schedule the jobs.

Let $\bar{x}$ and $\bar{z}$ be a solution of the Lagrangean subproblem (LP') and its objective function value respectively. Then γ are the Lagrangean multipliers obtained by the Volume Algorithm and $v = c - Bx$ is the subgradient at γ and x. Let UB be the upper bound or feasible solution, the initial UB is generated initially by a greedy heuristic such as Lawler (1976). The β cost is defined by the values of $\bar{\gamma}$ and $\bar{z}$. The steps of the algorithm proposed can be summarized as:

Step 0: We start with a null vector $\bar{\gamma}$, $\beta^0 = T$ (Tardiness - original objective function) and solve LP' to obtain $\bar{x}$ and $\bar{z}$. Let $\bar{UB}$ be the feasible solution obtained by the greedy heuristic. Set that $\bar{\beta}$ is a null vector, $x^0 = \bar{x}$, $z^0 = \bar{z}$ and $k = 1$.

Step 1: Compute the subgradient $v^k = c - B.\bar{x}$ and $\gamma^k = \bar{\gamma} + sv^k$ for a step size given by 19 and as NDRC proposed. Compute β^k such as proposed in (15). Solve PL' with β^k and let x^t and z^t be the solutions obtained. Then $\bar{x} = \alpha x^k + (1-\alpha)\bar{x}$ and $\bar{\beta} = \alpha\beta^k + (1-\alpha)\bar{\beta}$, where α is a number between 0 and 1, and defined by convex combination such as defined in (16).

Step 2: If $z^k > \bar{z}$ update $\bar{\gamma} = \gamma^k$ and $\bar{z} = z^k$. If $\bar{z}$ improves by 10% since the last run of the Lagrangean metaheuristic then go to Step 3. Else let $k = k + 1$ and go to Step 1.

Step 3: Lagrangean metaheuristic:

(i) list $\leftarrow$ sort all jobs in decreasing order of associated $\bar{\beta}$ values;

(ii) $UB^k \leftarrow$ NEH (list);

(iii) $UB^k \leftarrow$ ILS (UB^k,$\bar{\beta}$).

If $UB^k < UB$ update $UB = UB^k$. Let $k = k + 1$ and go to Step 1.

The proposed framework is sketched in Figure 1.

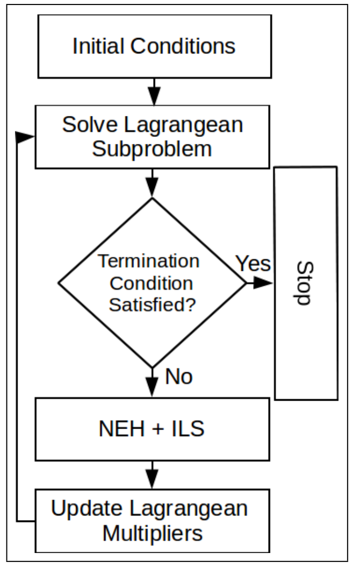


Figure 1: Hybrid Algorithm Proposed Diagram

The step size s and the parameter α used to define the $\bar{x}$ and $\bar{\beta}$, are computed as proposed in Barahona and Anbil (2000) and Fukuda (2007). First, we define α as:

$$\alpha_{opt} = argmin \parallel \alpha v'^k + (1-\alpha)\bar{v}' \parallel^2 \tag{16}$$

where, the subgradients in x^k and $\bar{x}$ can be defined as

$$v'^k = c - B.x^k,$$

$$\bar{v}' = c - B.\bar{x}.$$

The parameter α_{max} is initially defined as 0.25, based on computational tests presented in the Appendix A, and α is computed as:

$$\alpha = \frac{\alpha_{max}}{10} \text{ if } \alpha \leq 0, \tag{17}$$

$$\alpha = \min\{\alpha_{opt}, \alpha_{max}\} \text{ if } \alpha \geq 0. \tag{18}$$

The parameter α_{max} may decrease its value near the end, as this should increase the precision of the primal solution Fukuda (2007). After $n/10$ iterations, if z improves less than 1% and $\alpha_{max} \geq 0.00001$ then α_{max} is updated, $\alpha_{max} = 0.5\alpha_{max}$.

Before the definition of step size s, we need to define the parameter $\lambda \in [0.0005, 2]$, initially set to 0.1 based on computational tests presented in the Appendix A. Therefore, to set the value of λ we define three types of iterations based on Barahona and Anbil (2000) and Barahona and Ladányi (2006). The number of iterations for each type is defined based on computational tests presented in the Appendix A. Each iteration with no improvement is defined as *red*. If $z^k > \bar{z}$ and $v^k(c - B.x^k) < 0$, such as mentioned by Barahona and Anbil (2000), it means that a longer step in the direction v^k would have given a smaller value for z^k. Those iterations are denominated *yellow*, otherwise, the iteration is denominated *green*. At each *green* iteration we would multiply λ by 2, if the result is greater than 2, we set $\lambda = 2$. After two consecutive *yellow* iterations we also multiply λ by 1.1 and we set it to $\min\{\lambda, 2\}$. After a sequence of 30 consecutive *red* iterations, we would multiply λ by 0.66 and we set it to $\max\{\lambda, 0.0005\}$. Thus, the step size s at iteration k is defined as:

$$s = \frac{\lambda(UB - \bar{z})}{\| v^k \|^2}. \tag{19}$$

Finally, the stop criterion of the Volume algorithm is determined if one of the following criteria is satisfied: (*i*) Maximum number of iterations, in this case $10n$, where n is the number of jobs; (*ii*) Relative tolerance norm of the gradient vector v, defined as $\frac{c-B\bar{x}}{m} \leq 0.01$, where m the number of rows in the matrix; (*iii*) Relative tolerance difference between the lower bound ($\bar{z}$) and its primal function ($c\bar{x}$), defined as $\frac{\bar{z}-c\bar{x}}{\bar{z}} \leq 0.01$; (*iv*) Relative tolerance GAP defined as $\frac{UB-\bar{z}}{UB} \leq 0.01$.

2.5. *Nawaz-Enscore-Ham (NEH)*

The NEH heuristic was first proposed for the Permutation Flow-Shop Scheduling Problem by Nawaz et al. (1983), and then used by de Paula et al. (2010) for a parallel machine scheduling problem. It attempts to obtain good quality solutions by inserting jobs sequentially, in descending order of $\bar{\beta}$, at the most interesting position of the partial schedule.

The proposed NEH heuristic initially uses a "list" $\mathscr{L}$ with n jobs in descending order of $\bar{\beta}$. At each step, the job at the beginning of list is inserted in the best position, according to the objective function, Algorithm 2 describes the procedure.

2.6. *Iterated Local Search*

The Iterated Local Search (ILS), is a simple and effective metaheuristic, which iteratively applies local search in a singular way (Lourenço et al. (2003) and Naderi et al. (2010)). Stutzle (1998), Stutzle et al. (2001) and Hoos and Stützle

Algorithm 2 NEH - $O(n^2)$

Require: A sorted list $\mathscr{L}$ of n jobs
Ensure: A feasible solution with value UB
  for $j = 1$ to n **do**
    BestPosition $\leftarrow 0$
    BestObjetive $\leftarrow \infty$
    for Each Position l available **do**
      ObjectiveValue $\leftarrow$ objective function of the partial solution assuming that j is scheduled at position l
      if ObjectiveValue $<$ BestObjective **then**
        BestPosition $\leftarrow l$
        BestObjective $\leftarrow$ ObjectiveValue
      end if
    end for
    Schedule $\mathscr{L}[j]$ at position BestPosition processing queue
  end for
  return Solution $\mathscr{S}$

(2004) emphasize that the ILS is one of the simplest methods based on local search. It aims at preventing stagnation around a local optimum. The idea was originated in Baxter (1981) and cited by several authors, such as "iterated descent" (Baum (1986a,b)), "iterated Lin Kernighan" (Johnson (1990)), "large-step Markov chains" (Martin et al. (1991)), "chained local optimization" (Martin and Otto (1996)), among others (see Lourenço et al. (2003)). The ILS efficiency is highlighted in Johnson (1990), Martin et al. (1991), Brucker et al. (1996), Martin and Otto (1996), Brucker et al. (1997), Johnson and McGeoch (1997), Applegate et al. (2003), Lourenço et al. (2003), Rodrigues et al. (2008) and Lourenço et al. (2010).

The jobs with larger values of $\bar{\beta}$ have more probability of improving the solution by their change of position. In the proposed implementation we use the Lagrangean multipliers as guides to perturb the current solution. The main loop of the ILS has three main procedures which are applied iteratively until a stopping criterion is accepted (see Algorithm 3.) In this work, we determine the number without improvement as stopping criterion. Previous computational tests, presented in the Appendix A, showed that for $10n$ ($ILS_{max} = 10n$), n being the number of jobs, the best results are obtained.

The steps of the metaheuristic ILS are minutely described below:

Algorithm 3 ILS

Require: $\mathscr{S}$, $\bar{\beta}$, ILS_{max}
Ensure: A feasible solution with value UB
  while Iteration number without improvement $\leq ILS_{max}$ **do**
    $\mathscr{S}_1 \leftarrow$ *Perturbation*($\mathscr{S}$,$\bar{\beta}$)
    $\mathscr{S}_2 \leftarrow$ *LocalSearch*($\mathscr{S}_1$)
    $\mathscr{S} \leftarrow$ *AcceptanceCriterion*($\mathscr{S}$,$\mathscr{S}_2$)
  end while
  return Solution $\mathscr{S}$

(i) Perturbation: The first procedure is defined as "perturbation" and it consists in exchanging the actual solution in order to leave the local optimum. Based on the proposals of Stutzle (1998) and Arroyo et al. (2009), the perturbation consists in carrying out l swap moves - swapping of two adjacent jobs randomly chosen and $ceil(l/2)$ exchange moves - choosing two jobs, not necessarily adjacent, and exchanging them. In the proposed perturbation procedure, we define the random choices of jobs taking into account the costs $\bar{\beta}$ associated with each one, so that the jobs have probability to change their positions proportionally to $\bar{\beta}$ associated. In order to avoid a severe disruption in the exchange move, we make sure that the two jobs chosen for exchange are less than φ positions apart. After several computational tests, presented in the Appendix A, the best results are obtained with $l \in \{2,6\}$ and $\varphi \in \{0.25n, n\}$. The procedure of variation of l defines that each iteration of the ILS without improvement in local optimum, increases the number l of random changes in one unit. This number is increased until it reaches its upper bound. If the solution is improved, the value of l returns to its lower bound.

(ii) Local Search: The second procedure is the "local search", implemented and based on Den Besten et al. (2001), de Paula et al. (2007), Arroyo et al. (2009), de Paula et al. (2010) and Lourenço et al. (2010). In Den Besten et al. (2001), for single machine total weighted tardiness problem, and de Paula et al. (2007), is demonstrated with computational tests that swap moves before insertion moves presents better results. In addition, Den Besten et al. (2001) highlights that insertion before swaps moves performs only slightly worse. This fact is verified in Appendix A. Therefore, the adopted procedure is composed by "swaps" and "insertions" moves. The former

consists of interchanging all pairs of jobs. The latter consists of removing a job from its original position and inserting it on one of the $n-1$ remaining positions. The local search stops after it begins to be unable to improve the solution.

(iii) Acceptance Criterion: The acceptance criterion admits or discards the new local optimum. The new solution is accepted if its objective function value is better than the previous. Otherwise, it is discarded. Based on Congram et al. (2002) and Lourenço et al. (2010), we set up a procedure for backtracking the search, i.e., for every ϕ iterations of the ILS, the search solution returns to the initial solution provided by the NEH heuristic. According to previous tests, presented in the Appendix A, the best results are obtained for $\phi = 10$.

3. Computational Results

The benchmark used is from the study of Rubin and Ragatz (1995), available on-line at `https://www.msu.edu/~rubin/files/c&ordata.zip`. The scheduling problem can be classified as $1|d_j, s_{ij}|\sum_j T_j$ (Graham et al. (1979)). This set of problems is composed by four distinct sizes of jobs, (n = 15, 25, 35 and 45). For each number of jobs, eight variations of problems are generated, totaling 32 instances. The processing times are normally distributed with an average of 100 time units. The setup times are uniformly distributed with an average of 9.5 units of time. Each instance has three factors which posses two levels, high or low. These factors are the processing time variance, the tardiness and the range of due dates in the job sequence. The tardiness factor is roughly equivalent to the expected percentage of jobs in a randomly generated sequence that would be tardy. The experiments are run in the Linux Maya with single thread of 2.4 GHz and 4GB of ram memory.

The computational results are divided into two experiments. The first experiment compares the performance of the proposed hybrid algorithm (Hybrid Lagrangean Metaheuristic - HLM) with the heuristics which compose it. The second experiment compares the results obtained by the HLM proposed with previous algorithms from the literature which use the same set of instances, namely, “Iterated Local Search (ILS)” from Arroyo et al. (2009), “Greedy Randomized Adaptive Search Procedure (GRASP)” from Gupta and Smith (2006) and “Ant Colonies (AC)” from Liao and Juan (2007). The comparison of results and the design of tables are in accordance with Rardin and Uzsoy (2001). It is highlighted that in every experiment each instance is executed 5 times with independent random seeds.

Table 2 depicts the results of the first experiment. The table compares the performances of the proposed Hybrid Lagrangean Metaheuristic (*HLM*) with the ILS heuristic implemented with NEH (*ILSi*) and with the Volume Algorithm with NEH (*VNi*). Basically, *ILSi* consists of the ILS implemented with the initial solution generated by the NEH. *VNi* is the volume algorithm and it uses information from the Lagrangean multipliers and NEH to construct a feasible solution, as used in *HLM*.

Table 3 compares results of (*HLM*) with classic metaheuristics from literature, which use the same analyzed instances. The compared metaheuristics are, namely, GRASP, proposed by Gupta and Smith (2006) (*GRASPg*), AC proposed by Liao and Juan (2007) (*ACl*) and the ILS proposed by Arroyo et al. (2009) (*ILSa*). This table compares only the best results obtained, because the papers of Liao and Juan (2007) and Arroyo et al. (2009) do not report all information, which limited a wider comparison through literature. Readers must remember that each experiment was performed using a different computer architecture.

Table 2: Comparison of gaps between the Hybrid Lagrangean Metaheuristic Proposed (*HLM*) with implemented heuristics ILS (*ILSi*) and Volume Algorithm - NEH (*VNi*). The field “Instances” indicates the instance problem, and between brackets, its number of jobs. The field “Optimal” presents the optimal solution known of each problem and the field “$T_{best}(s)$” is the time in seconds to obtain the best result and this is restricted to 600 seconds. The results of the optimality relative gaps are distributed in W - Worst gap, M - Median gap and B - Best gap for each instance and each heuristic. It is highlighted in bold, in each row of the table, the heuristic that achieved the best computational result.

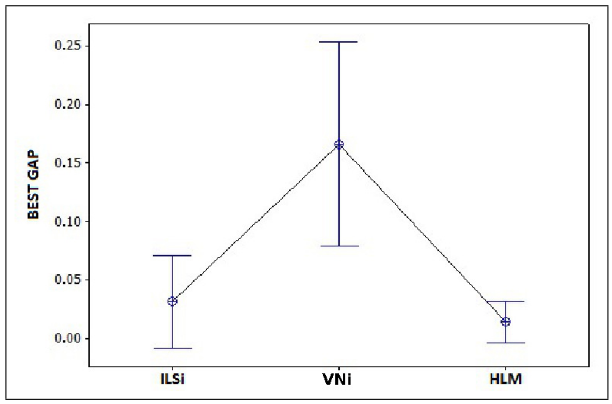
Instances		**ILSi**				**VNi**				**HLM**			
Problem	Optimal	W	M	B	$T_{best}(s)$	W	M	B	$T_{best}(s)$	W	M	B	$T_{best}(s)$
401(15)	90	3.23%	**0.00%**	**0.00%**	0.0	67.39%	64.84%	64.84%	61.2	**0.00%**	**0.00%**	**0.00%**	65.1
402(15)	0	**0.00%**	**0.00%**	**0.00%**	0.0	**0.00%**	**0.00%**	**0.00%**	65.0	**0.00%**	**0.00%**	**0.00%**	68.4
403(15)	3418	**0.00%**	**0.00%**	**0.00%**	0.0	10.48%	5.11%	5.11%	6.3	**0.00%**	**0.00%**	**0.00%**	7.3
404(15)	1067	**0.00%**	**0.00%**	**0.00%**	0.0	16.96%	11.96%	11.38%	11.4	**0.00%**	**0.00%**	**0.00%**	12.2
405(15)	0	**0.00%**	**0.00%**	**0.00%**	0.0	**0.00%**	**0.00%**	**0.00%**	0.9	**0.00%**	**0.00%**	**0.00%**	1.0
406(15)	0	**0.00%**	**0.00%**	**0.00%**	0.0	**0.00%**	**0.00%**	**0.00%**	54.8	**0.00%**	**0.00%**	**0.00%**	58.9
407(15)	1861	**0.00%**	**0.00%**	**0.00%**	0.0	28.34%	22.62%	19.26%	5.3	**0.00%**	**0.00%**	**0.00%**	5.9
408(15)	5660	**0.00%**	**0.00%**	**0.00%**	0.0	10.88%	3.76%	3.76%	41.6	**0.00%**	**0.00%**	**0.00%**	47.8
501(25)	261	1.88%	1.51%	0.76%	0.0	47.06%	43.87%	31.13%	99.8	**1.51%**	**0.76%**	**0.00%**	108.5
502(25)	0	**0.00%**	**0.00%**	**0.00%**	0.0	**0.00%**	**0.00%**	**0.00%**	154.1	**0.00%**	**0.00%**	**0.00%**	181.3
503(25)	3497	0.26%	**0.00%**	**0.00%**	0.0	12.53%	5.54%	10.01%	131.7	**0.00%**	**0.00%**	**0.00%**	147.9
504(25)	0	**0.00%**	**0.00%**	**0.00%**	0.0	**0.00%**	**0.00%**	**0.00%**	45.7	**0.00%**	**0.00%**	**0.00%**	50.3
505(25)	0	**0.00%**	**0.00%**	**0.00%**	0.0	**0.00%**	**0.00%**	**0.00%**	9.4	**0.00%**	**0.00%**	**0.00%**	10.2
506(25)	0	**0.00%**	**0.00%**	**0.00%**	0.0	**0.00%**	**0.00%**	**0.00%**	74.0	**0.00%**	**0.00%**	**0.00%**	86.1
507(25)	7225	**0.00%**	**0.00%**	**0.00%**	0.0	22.97%	18.34%	11.30%	66.2	**0.00%**	**0.00%**	**0.00%**	73.6
508(25)	1915	**0.00%**	**0.00%**	**0.00%**	0.0	55.04%	51.88%	36.88%	121.7	**0.00%**	**0.00%**	**0.00%**	129.5
601(35)	12	64.71%	61.29%	58.62%	4.0	88.46%	87.50%	86.36%	386.4	**45.45%**	**25.00%**	**25.00%**	429.3
602(35)	0	**0.00%**	**0.00%**	**0.00%**	0.0	**0.00%**	**0.00%**	**0.00%**	292.5	**0.00%**	**0.00%**	**0.00%**	328.6
603(35)	17587	0.50%	0.46%	0.21%	2.0	22.86%	16.69%	9.22%	398.8	**0.27%**	**0.22%**	**0.10%**	453.2
604(35)	19092	0.65%	0.22%	0.05%	7.0	12.25%	5.23%	3.37%	48.3	**0.22%**	**0.05%**	**0.00%**	51.9
605(35)	228	17.09%	11.63%	10.59%	2.0	58.70%	55.81%	44.12%	411.7	**7.32%**	**4.20%**	**1.72%**	442.7
606(35)	0	**0.00%**	**0.00%**	**0.00%**	0.0	**0.00%**	**0.00%**	**0.00%**	50.0	**0.00%**	**0.00%**	**0.00%**	53.2
607(35)	12969	0.29%	0.25%	0.00%	2.0	21.93%	17.25%	10.09%	124.3	**0.00%**	**0.00%**	**0.00%**	146.2
608(35)	4732	**0.00%**	**0.00%**	**0.00%**	2.0	23.75%	19.93%	18.48%	88.6	**0.00%**	**0.00%**	**0.00%**	95.3
701(45)	97	19.83%	18.49%	12.61%	14.0	80.28%	78.73%	76.28%	514.3	**13.39%**	**12.61%**	**11.01%**	565.2
702(45)	0	**0.00%**	**0.00%**	**0.00%**	0.0	**0.00%**	**0.00%**	**0.00%**	411.2	**0.00%**	**0.00%**	**0.00%**	437.4
703(45)	26506	0.71%	0.58%	0.32%	22.0	20.37%	14.80%	13.02%	396.5	**0.10%**	**0.07%**	**0.00%**	455.8
704(45)	15206	0.23%	**0.05%**	**0.00%**	20.0	12.09%	6.81%	5.66%	529.1	**0.05%**	**0.05%**	0.05%	562.9
705(45)	200	22.48%	18.70%	17.36%	28.0	65.99%	62.12%	61.39%	428.4	**11.50%**	**10.71%**	**9.09%**	442.6
706(45)	0	**0.00%**	**0.00%**	**0.00%**	0.0	**0.00%**	**0.00%**	**0.00%**	0.0	**0.00%**	**0.00%**	**0.00%**	69.6
707(45)	23789	0.51%	0.68%	0.51%	26.0	15.14%	8.35%	7.55%	517.5	**0.34%**	**0.24%**	**0.16%**	556.4
708(45)	22807	0.29%	0.08%	0.04%	28.0	10.73%	5.37%	4.57%	559.9	**0.04%**	**0.04%**	**0.00%**	589.3
Average		4.15%	3.56%	3.16%	4.9	22.01%	18.95%	16.68%	190.8	2.51%	1.69%	1.47%	210.4
Standard Deviation		12.50%	11.61%	10.91%	9.2	26.02%	25.77%	24.24%	192.8	8.49%	5.16%	4.96%	206.3

The algorithms presented varied results in Table 2, mainly between the average values of the best gaps (B), 3.16% for *ILSi*, 16.68% for *VNi* and 1.47% *HLM*. The average computational times showed a considerable variation from *ILSi* to *VNi* and *HLM*, with values of 4.9, 190.8 and 210.4 seconds, respectively. Analyzing individually the best result of gap (B), the hybrid metaheuristic, *HLM*, presented results equal to or smaller than the others, (except for the problem 704), and it achieved the optimal solution in 25 of the 32 instances consulted (78%). However and as expected, its computational time is higher, especially when compared to *ILSi*. The median result (M), and the worst result (W) of gaps, present the same relation as the discussed best gaps result (B). Although, *VNi* is wider from (W) to (B) than *ILSi* and *HLM*.

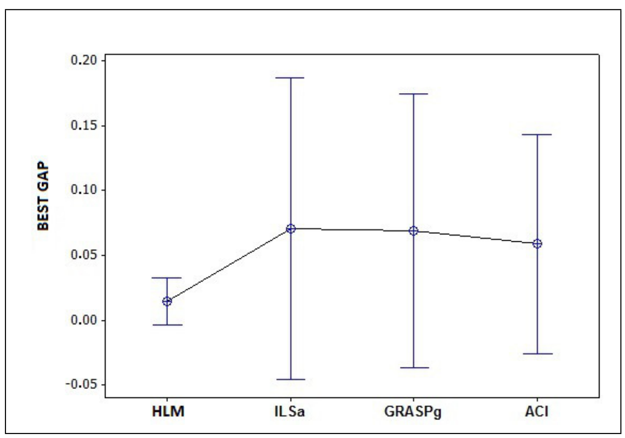
Due to the random components from the heuristics, the previous analysis may not be valid for every instance and/or moment. Thus, the ANOVA test is applied with significance level of 5% for the best gaps (B) in order to aver the statistical difference between the discussed heuristics. This test, performed with data of best gaps, obtained $p-value = 0.00008$. Therefore, the null hypothesis of equality of their averages, $p-value < 0.05$ for a significance level of 5% it is rejected . Thus, the analyzed heuristics present significant difference.

However, ANOVA does not provide enough information about which heuristics are responsible for this difference. To perform this multiple comparisons, all pairs of averages must be compared two by two. Tukey's test and a significance level of 5% for the best gaps (B) is used for this purpose. It obtained the following $p-values$ for the pairs: $VNi-ILSi$ of 0.001, $HLM-ILSi$ of 0.87 and $HLM-VNi$ of 0.0002. It is evidenced with the obtained values that there is a significant difference ($p-value < 0.05$) among the best gaps of *HLM* and *VNi*, *ILSi* and *VNi*, even though there is not statistical difference between *HLM* and *ILSi*. It is important to remember that *HLM*, provides lower and upper bounds in the same procedure, i.e., it is the only of them able to evaluate optimality conditions.

Another analysis is exposed in Figure (2a), which compares the best gaps 95% Confidence Interval (CI) for the analyzed heuristics. The heuristics *ILSi*, *VNi* and *HLM* presented the respective best gaps values for the 95% CI: $(-0.008; 0.071)$, $(0.079; 0.253)$ and $(-0.003; 0.032)$. Although Tukey's test results did not present a significant difference among all of the heuristics, the Figure (2a) shows that the *HLM* (Hybrid Lagrangean Metaheuristic Proposed) has a smaller level and amplitude of CI for the best gaps. Confidence intervals for *VNi* appears to be very different when compared to the other algorithms, presenting larger level and amplitude for the best gap (B). The use of the hybrid approach seems to be very effective, taking advantages from the combination of algorithms.



(a) Implemented Heuristics



(b) Implemented Heuristic Hybrid and Literature Heuristics

Figure 2: Comparison of Best Gaps 95% Confidence Interval for the Heuristics Analyzed

Table 3: Comparison of best gaps (*GAP*) for the Hybrid Lagrangean Metaheuristic Proposed (*HLM*) with classic metaheuristics results of literature, the GRASP of Gupta and Smith (2006) (*GRASPg*), the AC of Liao and Juan (2007) (*ACl*) and the ILS of Arroyo et al. (2009) (*ILSa*). The field "Instances" indicates the instance problem, and between brackets, its number of jobs. The field "Optimal" presents the optimal solution known of each problem and the field "$T_{best}(s)$" is the time in seconds to obtain the best result and this is restricted to 600 seconds. This table compares only the best results obtained, because the papers of Liao and Juan (2007) and Arroyo et al. (2009) present incomplete information to perform a wider comparison through literature. It is highlighted in bold, in each row of the table, the heuristic that achieved the best computational result.

Instances		**HLM**		**ILSa**		**GRASPg**		**ACl**
Problem	Optimal	*GAP*	$T_{best}(s)$	*GAP*	$T_{best}(s)$	*GAP*	$T_{best}(s)$	*GAP*
401(15)	90	**0.00%**	65.1	**0.00%**	0.3	**0.00%**	4.0	**0.00%**
402(15)	0	**0.00%**	68.4	**0.00%**	0.0	**0.00%**	0.0	**0.00%**
403(15)	3418	**0.00%**	7.3	**0.00%**	0.2	**0.00%**	3.7	**0.00%**
404(15)	1067	**0.00%**	12.2	**0.00%**	0.1	**0.00%**	2.5	**0.00%**
405(15)	0	**0.00%**	1.0	**0.00%**	0.0	**0.00%**	0.0	**0.00%**
406(15)	0	**0.00%**	58.9	**0.00%**	0.0	**0.00%**	0.0	**0.00%**
407(15)	1861	**0.00%**	5.9	**0.00%**	0.2	**0.00%**	3.9	**0.00%**
408(15)	5660	**0.00%**	47.8	**0.00%**	0.2	**0.00%**	3.2	**0.00%**
501(25)	261	**0.00%**	108.5	1.53%	1.0	0.77%	14.0	1.53%
502(25)	0	**0.00%**	181.3	**0.00%**	0.0	**0.00%**	0.0	**0.00%**
503(25)	3497	**0.00%**	147.9	**0.00%**	1.2	**0.00%**	18.5	**0.00%**
504(25)	0	**0.00%**	50.3	**0.00%**	0.0	**0.00%**	0.0	**0.00%**
505(25)	0	**0.00%**	10.2	**0.00%**	0.0	**0.00%**	0.0	**0.00%**
506(25)	0	**0.00%**	86.1	**0.00%**	0.0	**0.00%**	0.0	**0.00%**
507(25)	7225	**0.00%**	73.6	**0.00%**	1.3	**0.00%**	24.4	**0.00%**
508(25)	1915	**0.00%**	129.5	12.53%	1.6	12.53%	23.3	12.53%
601(35)	12	**25.00%**	429.3	183.33%	3.8	166.67%	53.3	133.33%
602(35)	0	**0.00%**	328.6	**0.00%**	0.0	**0.00%**	0.0	**0.00%**
603(35)	17587	**0.10%**	453.2	3.99%	3.5	3.37%	94.4	3.78%
604(35)	19092	**0.00%**	51.9	0.87%	3.1	0.98%	88.8	0.87%
605(35)	228	1.72%	442.7	**0.00%**	3.6	6.58%	59.0	5.26%
606(35)	0	**0.00%**	53.2	**0.00%**	0.0	**0.00%**	0.0	**0.00%**
607(35)	12969	**0.00%**	146.2	0.49%	3.7	0.49%	88.8	0.79%
608(35)	4732	**0.00%**	95.3	5.47%	4.2	5.47%	83.5	5.47%
701(45)	97	11.01%	565.2	11.34%	7.2	**10.31%**	122.5	**10.31%**
702(45)	0	**0.00%**	437.4	**0.00%**	0.0	**0.00%**	0.0	**0.00%**
703(45)	26506	**0.00%**	455.8	0.24%	7.2	0.54%	216.4	0.34%
704(45)	15206	**0.05%**	562.9	0.64%	6.6	0.64%	201.3	2.01%
705(45)	200	9.09%	442.6	**4.00%**	6.9	11.00%	129.9	9.50%
706(45)	0	**0.00%**	69.6	**0.00%**	0.0	**0.00%**	0.0	**0.00%**
707(45)	23789	**0.16%**	556.4	1.31%	6.8	1.41%	253.3	1.95%
708(45)	22807	**0.00%**	589.3	0.83%	7.7	0.83%	267.0	1.90%
Average		1.47 %	210.4	7.08%	2.2	6.92%	54.9	5.92%
Standard Deviation		4.96 %	206.3	32.31%	2.7	29.36%	79.9	23.48%

All analyzed heuristics in Table 3 present "best gaps" (B) with values lower than 8%. The heuristic *ILSa* presented the shortest computational time, 2.2 seconds, (the computational times of *ACl* are not reported by Liao and Juan (2007)), and the gap smaller than others for problems 605 and 705. The hybrid metaheuristic *HLM* presented average "best gap" results smaller than or equal to the ones presented by the other algorithms in 29 of 32 instances (91%), proving the optimal solution in 25 instances. Again, as expected, it presents the higher computational time, 210.4 seconds. *GRASPg* presented similar results to *ILSa*, 6.92% and 7.08%, respectively. Meanwhile, its computational time is quite superior, 54.9 and 2.2 seconds respectively. Lastly, *ACl* presented the smaller average best gap among the heuristics consulted in literature. *ACl* and *GRASPg* showed better results than *HLM* only in problem 701. It is worth remembering that *HLM* is the only algorithm in the comparison able to prove optimality of the instances.

Once more, the ANOVA test is applied for the "best gaps" with significance level of 5%. It is obtained $p - value = 0.78$, indicating that there is no statistical significant difference among the analyzed heuristics. However, with analysis of Figure (2b), the following values of 95% confidence interval (CI) for the best gaps are obtained: $(-0.003; 0.032)$, $(-0.046; 0.187)$, $(-0.037; 0.175)$ and $(-0.025; 0.144)$, for heuristics *HLM*, *ILSa*, *GRASPg* and *ACl*, respectively. Even though there is no significant difference among the heuristics by the ANOVA test, one may note that *HLM* presented the smallest amplitude of CI and values nearer to zero (null gap). In the same manner, it is verified that *ACl* presented smaller amplitude of CI and level when compared to algorithms, *ILSa* and *GRASPg*. These last present great similarity between their results.

4. Concluding remarks

The Hybrid Lagrangean Metaheuristic proposed in this manuscript is applied to the single machine scheduling problem with setup times and due dates to minimize total tardiness. This algorithm uses information from the objective function and from the Lagrangean multipliers in combination to two well known heuristics NEH and the ILS, to obtain a refined solution with reduced computational time. The mathematical model used by the Lagrangean relaxation was the one formulated by Sousa and Wolsey (1992).

The proposed algorithm (*HLM*) presents it self very competitive with classic metaheuristics of great performance, GRASP from Gupta and Smith (2006), AC from Liao and Juan (2007) and ILS from Arroyo et al. (2009). Furthermore, it is the only of them able to evaluate optimality conditions. Among the 32 problems tested, the (*HLM*) proved optimality in 25 of them.

The higher computational time of the proposed hybrid algorithm is expected, as the formulation used for the Lagrangean relaxation procedure presents $O(n^2h)$ constraints, which means, the number of iterations of the Volume algorithm will be impacted by n^2 and h (see Nogueira et al. (2014)). This fact does not occur in the other algorithms (GRASP, ILS and AC), which only depend on n. Therefore, as $h >>> n$ the time increases faster in algorithms *HLM* and *VNi*. The imposed time limit of 600 seconds cause a decay in the performance of *HLM* and *VNi* as the instances get bigger. To avoid the impact of solving the Lagrangean subproblem for bigger instances, a novel strategy is been analyzed as future research, which consists in interrupting the Volume algorithm as soon as a lower bound stability objective is reached.

Appendix A. Parameters Tuning

For parameters tuning, five independent instances for four distinct sizes of jobs, n = 15, 25, 35 and 45 are artificially created based on the works of Rubin and Ragatz (1995) and Nogueira et al. (2014). Thus, 20 instances are randomly and independently generated. The processing times are normally distributed with an average of 100 time units and the setup times are uniformly distributed with an average of 9.5 units of time. The due dates are uniformly distributed between $\max_j\{p_j\}$ and $\frac{h'}{2}$, where $h' = \sum_j p_j + \sum_i \max_j\{s_{ij}\}$. For each parameter of each method, all instances are solved individually changing its parameters from lower to upper defined limits. The parameters are chosen through the analysis of the average gap. For the initial tests the parameters are based in literature benchmarking. In each test of each parameter analyzed, just one parameter is changed from literature benchmarking, i.e., others are kept fixed. For Volume Algorithm for initial values the works of Barahona and Ladányi (2006) and Fukuda (2007) were considered and for ILS, the work of Arroyo et al. (2009) was considered. In presented parameters, for Volume Algorithm and for Iterated Local Search, the value between parenthesis is the chosen parameter value for this manuscript.

Appendix A.1. Volume Algorithm

The "Volume algorithm" tuning is based in the following parameters:

α_{max}: varying between 0.01 and 1, with literature value defined as 0.1 (0.25),

$\lambda_{initial}$: varying between 0.01 and 2, with literature value defined as 1 (0.1), *green*: varying between 1 and 10, with literature value defined as 1 (1),

yellow: varying between 1 and 10, with literature value defined as 2 (2),

red: varying between 1 and 40, with literature value defined as 20 (30).

Appendix A.2. Iterated Local Search

The "Iterated Local Search" algorithm tuning is based in the following parameters:

ILS_{max}: varying between n and $20n$, with literature value defined as 1000 ($10n$),

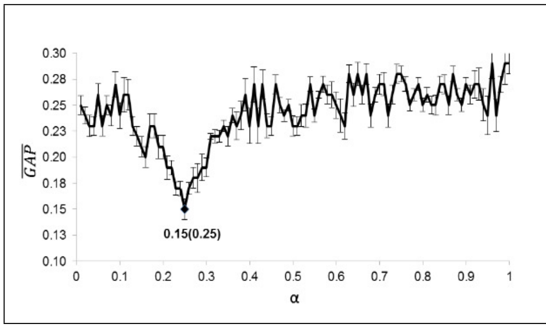
l: varying its upper limit between 1 and 10, with literature value defined as 4 (6),

φ: varying its lower limit between $0.05n$ and $0.75n$, with literature value defined as $0.20n$ ($0.25n$),

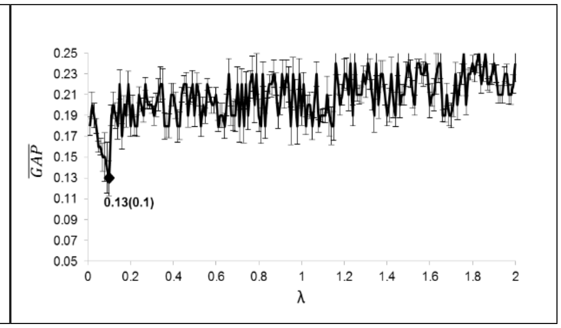
ϕ: varying between 1 and 25, with literature value defined as 5 (10).

Appendix A.3. Local Search

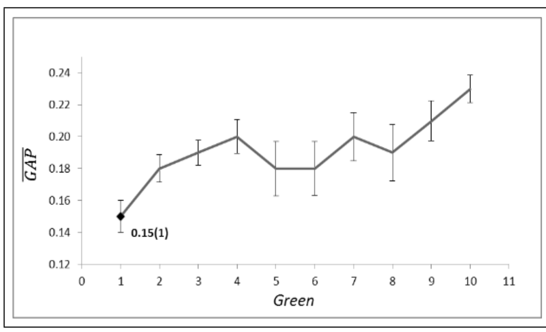
The "Local Search" strategy compares the performance of "Swap-Insertion" and "Insertion-Swap". The first implements the swap moves before insertion moves while the last does the opposite.



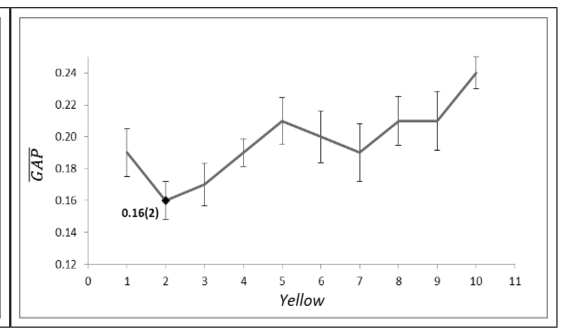
(a) Average GAP Results for $\alpha_m ax$ Tuning



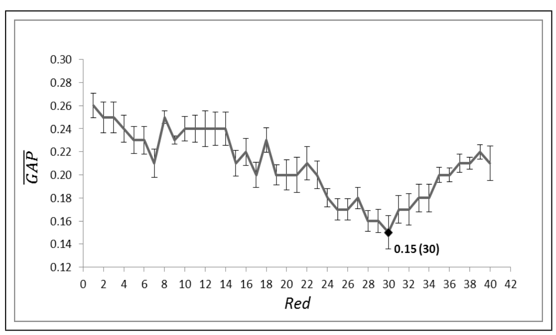
(b) Average GAP Results for $\lambda_{initial}$ Tuning



(c) Average GAP Results for *Green* Tuning

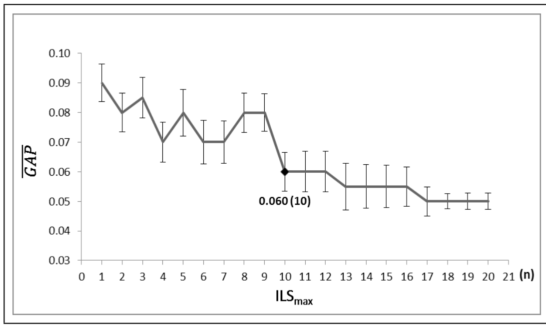


(d) Average GAP Results for *Yellow* Tuning

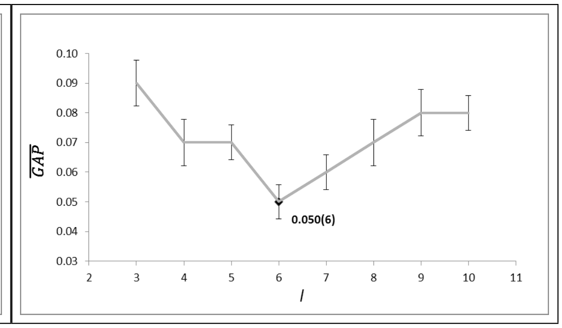


(e) Average GAP Results for *Red* Tuning

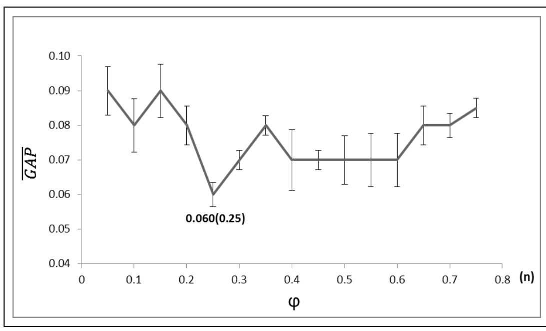
Figure A.3: Average Gaps 95% Confidence Interval Results for Algorithm Volume Parameter Tuning



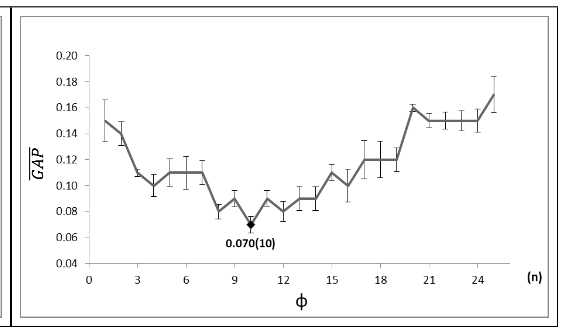
(a) Average GAP Results for ILS_{max} Tuning



(b) Average GAP Results for l Tuning



(c) Average GAP Results for φ Tuning



(d) Average GAP Results for ϕ Tuning

Figure A.4: Average Gaps 95% Confidence Interval for Iterated Local Search Parameter Tuning

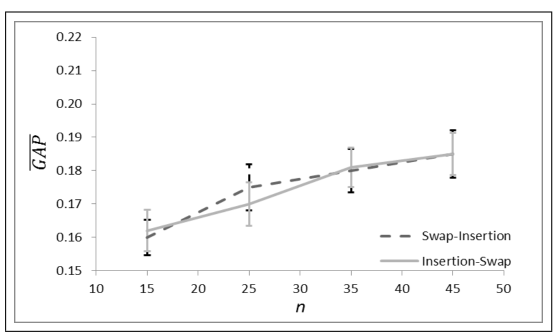


Figure A.5: Average Gaps 95% Confidence Interval Results for Analysis of the Local Search Strategy

Acknowledgements

M.G. Ravetti acknowledge support from CNPq and FAPEMIG, Brazil.

References

Alba, E.. Parallel metaheuristics: a new class of algorithms; vol. 47. Wiley. com; 2005.

Puchinger, J., Raidl, G.R.. Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. In: Artificial intelligence and knowledge engineering applications: a bioinspired approach. Springer; 2005, p. 41–53.

Raidl, G.R.. A unified view on hybrid metaheuristics. In: Hybrid Metaheuristics. Springer; 2006, p. 1–12.

Blum, C., Roli, A.. Hybrid metaheuristics: an introduction. In: Hybrid Metaheuristics. Springer; 2008, p. 1–30.

Raidl, G.R., Puchinger, J.. Combining (integer) linear programming techniques and metaheuristics for combinatorial optimization. In: Hybrid Metaheuristics. Springer; 2008, p. 31–62.

Jourdan, L., Basseur, M., Talbi, E.G.. Hybridizing exact methods and metaheuristics: A taxonomy. European Journal of Operational Research 2009;199(3):620–629.

Raidl, G.R., Puchinger, J., Blum, C.. Metaheuristic hybrids. In: Handbook of Metaheuristics. Springer; 2010, p. 469–496.

Blum, C., Puchinger, J., Raidl, G.R., Roli, A.. Hybrid metaheuristics in combinatorial optimization: A survey. Applied Soft Computing 2011;11(6):4135–4151.

Talbi, E.G.. A taxonomy of hybrid metaheuristics. Journal of heuristics 2002;8(5):541–564.

Dumitrescu, I., Stützle, T.. Combinations of local search and exact algorithms. In: Applications of Evolutionary Computing. Springer; 2003, p. 211–223.

Danna, E., Le Pape, C.. Two generic schemes for efficient and robust cooperative algorithms. In: Constraint and Integer Programming. Springer; 2004, p. 33–57.

Cotta, C., Talbi, E., Alba, E.. 1 parallel hybrid metaheuristics 2005;.

Ehrgott, M., Gandibleux, X.. Hybrid metaheuristics for multi-objective combinatorial optimization. In: Hybrid metaheuristics. Springer; 2008, p. 221–259.

Blum, C., Puchinger, J., Raidl, G., Roli, A., et al. A brief survey on hybrid metaheuristics. In: Proceedings of BIOMA. 2010, p. 3–18.

Colorni, A., Dorigo, M., Maniezzo, V., et al. Distributed optimization by ant colonies. In: Proceedings of the first European conference on artificial life; vol. 142. Paris, France; 1991, p. 134–142.

Steiglitz, K., Papadimitriou, C.H.. Combinatorial optimization: Algorithms and complexity. Prentice Hall, New Jersey, UV Vazirani (1984) On two geometric problems related to the travelling salesman problem J Algorithms 1982;5:231–246.

Holland, J.. Adaptation in Natural and Artificial Systems. Ann Arbor, MI, USA: Michigan Press University; 1975.

Lawler, E.L.. Combinatorial optimization: Networks and matroids. 1976; vol. 3. New York; 1976.

Feo, T.A., Resende, M.G.C.. Greedy randomized adaptive search procedures. Journal of Global Optimization 1995;6:109–133.

White, D.A., Sofge, D.A.. Handbook of Intelligent Control: Neural, Fuzzy, and Adaptative Approaches. Van Nostrand Reinhold Company; 1992.

Lourenço, H.R., Martin, O.C., Stutzle, T.. Iterated local search. In: Glover, F., Kochenberger, G., editors. Handbook of Metaheuristics; vol. 57 of *International Series in Operations Research & Management Science*. Springer US. ISBN 978-1-4020-7263-5; 2003, p. 320–353.

Kirkpatrick, S., Jr., D.G., Vecchi, M.P.. Optimization by simmulated annealing. science 1983;220(4598):671–680.

Glover, F.. Tabu search-part i. ORSA Journal on computing 1989;1(3):190–206.

Mladenović, N., Hansen, P.. Variable neighborhood search. Computers & Operations Research 1997;24(11):1097–1100.

Osman, I.H., Laporte, G.. Metaheuristics: A bibliography. Annals of Operations Research 1996;63(5):511–623.

Reeves, C.R.. Modern heuristic techniques for combinatorial problems. John Wiley & Sons, Inc.; 1993.

Nemhauser, G.L., Wolsey, L.A.. Integer and combinatorial optimization; vol. 18. Wiley New York; 1988.

Wolsey, L.A.. Mixed Integer Programming. Wiley Encyclopedia of Computer Science and Engineering; 2008.

Balakrishnan, J., Cheng, C.H., Conway, D.G., Lau, C.M.. A hybrid genetic algorithm for the dynamic plant layout problem. International Journal of Production Economics 2003;86(2):107–120.

Cortinhal, M.J., Captivo, M.E.. Upper and lower bounds for the single source capacitated location problem. European Journal of Operational Research 2003;151(2):333–351.

Cotta, C., Troya, J.M.. Embedding branch and bound within evolutionary algorithms. Applied Intelligence 2003;18(2):137–153.

Danna, E., Rothberg, E., Le Pape, C.. Integrating mixed integer programming and local search: A case study on job-shop scheduling problems. In: Fifth International Workshop on Integration of AI and OR techniques in Constraint Programming for Combinatorial Optimisation Problems (CP-AI-OR'2003). 2003, p. 65–79.

Greistorfer, P.. A tabu scatter search metaheuristic for the arc routing problem. Computers & Industrial Engineering 2003;44(2):249–266.

Haouari, M., Ladhari, T.. A branch-and-bound-based local search method for the flow shop problem. Journal of the Operational Research Society 2003;54(10):1076–1084.

Jahuira, C., Cuadros-Vargas, E.. Solving the tsp by mixing gas with minimal spanning trees. In: 1st International Conference of the Peruvian Computer Society. 2003, p. 123–132.

Klepeis, J., Pieja, M., Floudas, C.. Hybrid global optimization algorithms for protein structure prediction: Alternating hybrids. Biophysical journal 2003;84(2):869–882.

Neto, T., Pedroso, J.P.. Grasp for linear integer programming. Applied Optimization 2003;86:545–574.

Shi, H., Wang, L.. A mixed branch-and-bound and neural network approach for the broadcast scheduling problem. In: Design and application of hybrid intelligent systems. IOS Press; 2003, p. 42–49.

Spina, R., Galantucci, L.M., Dassisti, M.. A hybrid approach to the single line scheduling problem with multiple products and sequence-dependent time. Computers & industrial engineering 2003;45(4):573–583.

Alvim, A.C., Ribeiro, C.C., Glover, F., Aloise, D.J.. A hybrid improvement heuristic for the one-dimensional bin packing problem. Journal of Heuristics 2004;10(2):205–229.

Barake, M., Chardaire, P., McKeown, G.P.. The probe metaheuristic and its application to the multiconstraint knapsack problem. In: Metaheuristics. Kluwer Academic Publishers; 2004, p. 19–36.

Basseur, M., Lemesre, J., Dhaenens, C., Talbi, E.G.. Cooperation between branch and bound and evolutionary approaches to solve a bi-objective flow shop problem. In: Experimental and Efficient Algorithms. Springer; 2004, p. 72–86.

Della Croce, F., Ghirardi, M., Tadei, R.. Recovering beam search: Enhancing the beam search approach for combinatorial optimization problems. Journal of Heuristics 2004;10(1):89–104.

Feltl, H., Raidl, G.R.. An improved hybrid genetic algorithm for the generalized assignment problem. In: Proceedings of the 2004 ACM symposium on Applied computing. ACM; 2004, p. 990–995.

Focacci, F., Laburthe, F., Lodi, A.. Local search and constraint programming. In: Constraint and Integer Programming. Springer; 2004, p. 293–329.

Grosso, A., Della Croce, F., Tadei, R.. An enhanced dynasearch neighborhood for the single-machine total weighted tardiness scheduling problem. Operations Research Letters 2004;32(1):68–72.

Kanazawa, T., Yasuda, K.. Proximate optimality principle based tabu search. IEEJ Transactions on Electronics, Information and Systems 2004;124:912–920.

Klau, G.W., Ljubić, I., Moser, A., Mutzel, P., Neuner, P., Pferschy, U., et al. Combining a memetic algorithm with integer programming to solve the prize-collecting steiner tree problem. In: Genetic and Evolutionary Computation–GECCO 2004. Springer; 2004, p. 1304–1315.

Koduru, P., Das, S., Welch, S., Roe, J.L.. A multi-objective ga-simplex hybrid approach for gene regulatory network models. In: Evolutionary Computation, 2004. CEC2004. Congress on; vol. 2. IEEE; 2004, p. 2084–2091.

Kostikas, K., Fragakis, C.. Genetic programming applied to mixed integer programming. In: Genetic Programming. Springer; 2004, p. 113–124.

Lin, Z.Z., Bean, J.C., White, C.C.. A hybrid genetic/optimization algorithm for finite-horizon, partially observed markov decision processes. INFORMS Journal on Computing 2004;16(1):27–38.

Lozano, M., Herrera, F., Krasnogor, N., Molina, D.. Real-coded memetic algorithms with crossover hill-climbing. Evolutionary computation 2004;12(3):273–302.

Palpant, M., Artigues, C., Michelon, P.. Lssper: solving the resource-constrained project scheduling problem with large neighbourhood search. Annals of Operations Research 2004;131(1-4):237–257.

Perron, L., Shaw, P., Furnon, V.. Propagation guided large neighborhood search. In: Principles and Practice of Constraint Programming-CP 2004. Springer; 2004, p. 468–481.

Puchinger, J., Raidl, G.R.. An evolutionary algorithm for column generation in integer programming: an effective approach for 2d bin packing. In: Parallel Problem Solving from Nature-PPSN VIII. Springer; 2004, p. 642–651.

Angel, E., Bampis, E.. A multi-start dynasearch algorithm for the time dependent single-machine total weighted tardiness scheduling problem. European Journal of Operational Research 2005;162(1):281–289.

Blum, C.. Beam-aco–hybridizing ant colony optimization with beam search: An application to open shop scheduling. Computers & Operations Research 2005a;32(6):1565–1591.

Blum, C.. Ant colony optimization: Introduction and recent trends. Physics of Life reviews 2005b;2(4):353–373.

Chelouah, R., Siarry, P.. A hybrid method combining continuous tabu search and nelder–mead simplex algorithms for the global optimization of multiminima functions. European Journal of Operational Research 2005;161(3):636–654.

Cowling, P.I., Keuthen, R.. Embedded local search approaches for routing optimization. Computers & operations research 2005;32(3):465–490.

Danna, E., Rothberg, E., Le Pape, C.. Exploring relaxation induced neighborhoods to improve mip solutions. Mathematical Programming 2005;102(1):71–90.

Dunker, T., Radons, G., Westkämper, E.. Combining evolutionary computation and dynamic programming for solving a dynamic facility layout problem. European Journal of Operational Research 2005;165(1):55–69.

Gallardo, J.E., Cotta, C., Fernández, A.J.. Solving the multidimensional knapsack problem using an evolutionary algorithm hybridized with branch and bound. In: Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach. Springer; 2005, p. 21–30.

Imahori, S., Yagiura, M., Ibaraki, T.. Improved local search algorithms for the rectangle packing problem with general spatial costs. European Journal of Operational Research 2005;167(1):48–67.

Lichtenberger, D.. An extended local branching framework and its application to the multidimensional knapsack problem. Ph.D. thesis; Citeseer; 2005.

Nwana, V., Darby-Dowman, K., Mitra, G.. A co-operative parallel heuristic for mixed zero–one linear programming: Combining simulated annealing with branch and bound. European journal of operational research 2005;164(1):12–23.

Pedroso, J.P., Kubo, M.. Hybrid tabu search for lot sizing problems. In: Hybrid Metaheuristics. Springer; 2005, p. 66–77.

Prandtstetter, M., Raidl, G.R.. A variable neighborhood search approach for solving the car sequencing problem. In: Proceedings of the 18th Mini Euro Conference on Variable Neighborhood Search, Tenerife, Spain. 2005,.

Price, K., Storn, R.M., Lampinen, J.A.. Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series). Secaucus, NJ, USA: Springer-Verlag New York, Inc.; 2005. ISBN 3540209506.

Puchinger, J., Raidl, G.R., Gruber, M.. Cooperating memetic and branch-and-cut algorithms for solving the multidimensional knapsack problem. In: Proceedings of MIC2005, the 6th Metaheuristics International Conference. Vienna, Austria: Universität Wien; 2005, p. 775–780.

Rego, C.. Ramp: A new metaheuristic framework for combinatorial optimization. In: Metaheuristic Optimization via Memory and Evolution. Springer; 2005, p. 441–460.

Shi, X., Liang, Y., Lee, H., Lu, C., Wang, L.. An improved ga and a novel pso-ga-based hybrid algorithm. Information Processing Letters 2005;93(5):255–261.

Vasquez, M., Vimont, Y.. Improved results on the 0–1 multidimensional knapsack problem. European Journal of Operational Research 2005;165(1):70–81.

De Franceschi, R., Fischetti, M., Toth, P.. A new ilp-based refinement heuristic for vehicle routing problems. Mathematical Programming 2006;105(2-3):471–499.

Doerner, K.F., Gutjahr, W.J., Hartl, R.F., Strauss, C., Stummer, C.. Pareto ant colony optimization with ilp preprocessing in multiobjective project portfolio selection. European Journal of Operational Research 2006;171(3):830–841.

Glover, F.. Parametric tabu-search for mixed integer programs. Computers & Operations Research 2006;33(9):2449–2494.

Gomes, A.M., Oliveira, J.F.. Solving irregular strip packing problems by hybridising simulated annealing and linear programming. European Journal of Operational Research 2006;171(3):811–829.

Hansen, P., Mladenović, N., Urošević, D.. Variable neighborhood search and local branching. Computers & Operations Research 2006;33(10):3034–3045.

Haouari, M., Chaouachi Siala, J.. A hybrid lagrangian genetic algorithm for the prize collecting steiner tree problem. Computers & Operations Research 2006;33(5):1274–1288.

Ibaraki, T., Nakamura, K.. Packing problems with soft rectangles. In: Hybrid Metaheuristics. Springer; 2006, p. 13–27.

Puchinger, J., Raidl, G.R., Pferschy, U.. The core concept for the multidimensional knapsack problem. In: Evolutionary Computation in Combinatorial Optimization. Springer; 2006, p. 195–208.

Ropke, S., Pisinger, D.. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transportation science 2006;40(4):455–472.

Sniedovich, M., Voß, S.. The corridor method: a dynamic programming inspired metaheuristic. Control and Cybernetics 2006;35(3):551–578.

Stützle, T.. Iterated local search for the quadratic assignment problem. European Journal of Operational Research 2006;174(3):1519–1539.

Umetani, S., Yagiura, M., Ibaraki, T.. One-dimensional cutting stock problem with a given number of setups: a hybrid approach of metaheuristics and linear programming. Journal of Mathematical Modelling and Algorithms 2006;5(1):43–64.

Wilbaut, C., Hanafi, S., Fréville, A., Balev, S., et al. Tabu search: global intensification using dynamic programming. Control and Cybernetics 2006;35(3):579–598.

Beck, J.C.. Solution-guided multi-point constructive search for job shop scheduling. J Artif Intell Res(JAIR) 2007;29:49–77.

Blum, C., Cotta, C., Fernández, A.J., Gallardo, J.E.. A probabilistic beam search approach to the shortest common supersequence problem. In: Evolutionary Computation in Combinatorial Optimization. Springer; 2007, p. 36–47.

Chaves, A.A., de Assis Correa, F., Lorena, L.A.N.. Clustering search heuristic for the capacitated p-median problem. In: Innovations in Hybrid Intelligent Systems. Springer; 2007, p. 136–143.

Dahal, K.P., Aldridge, C.J., Galloway, S.J.. Evolutionary hybrid approaches for generation scheduling in power systems. European journal of operational research 2007;177(3):2050–2068.

Gallardo, J.E., Cotta, C., Fernández, A.J.. On the hybridization of memetic algorithms with branch-and-bound techniques. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 2007;37(1):77–83.

Gomes da Silva, C., Figueira, J., Clímaco, J.. Integrating partial optimization with scatter search for solving bi-criteria $\{0, 1\}$-knapsack problems. European Journal of Operational Research 2007;177(3):1656–1677.

Hansen, P., Brimberg, J., Urošević, D., Mladenović, N.. Primal-dual variable neighborhood search for the simple plant-location problem. INFORMS Journal on Computing 2007;19(4):552–564.

Jeet, V., Kutanoglu, E.. Lagrangian relaxation guided problem space search heuristics for generalized assignment problems. European journal of operational research 2007;182(3):1039–1056.

Kiziltan, Z., Lodi, A., Milano, M., Parisini, F.. Cp-based local branching. In: Principles and Practice of Constraint Programming–CP 2007. Springer; 2007, p. 847–855.

Öncan, T., Kabadi, S., Nair, K., Punnen, A.. Vlsn search algorithms for partitioning problems using matching neighbourhoods. Journal of the operational research society 2007;59(3):388–398.

Pirkwieser, S., Raidl, G.R., Puchinger, J.. Combining lagrangian decomposition with an evolutionary algorithm for the knapsack constrained maximum spanning tree problem. In: Evolutionary Computation in Combinatorial Optimization. Springer; 2007, p. 176–187.

Puchinger, J., Raidl, G.R.. Models and algorithms for three-stage two-dimensional bin packing. European Journal of Operational Research 2007;183(3):1304–1327.

Ravetti, M.G.. Algoritmos para o problema de sequenciamento com máquinas paralelas e tempos de preparação dependentes da sequência. Ph.D. thesis; Pós Graduação em Ciência da Computação/UFMG; 2007.

Reimann, M.. Guiding aco by problem relaxation: a case study on the symmetric tsp. In: Hybrid Metaheuristics. Springer; 2007, p. 45–56.

Rothberg, E.. An evolutionary algorithm for polishing mixed integer programming solutions. INFORMS Journal on Computing 2007;19(4):534–541.

Trick, M.A., Yildiz, H.. A large neighborhood search heuristic for graph coloring. In: Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. Springer; 2007, p. 346–360.

Tse, S.M., Liang, Y., Leung, K.S., Lee, K.H., Mok, T.K.. A memetic algorithm for multiple-drug cancer chemotherapy schedule optimization. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 2007;37(1):84–91.

Blum, C.. Beam-aco for simple assembly line balancing. INFORMS Journal on Computing 2008;20(4):618–627.

Blum, C., Cotta, C., Fernández, A.J., Gallardo, J.E., Mastrolilli, M.. Hybridizations of metaheuristics with branch & bound derivates. In: Hybrid Metaheuristics. Springer; 2008, p. 85–116.

Chen, Y.W., Lu, Y.Z., Yang, G.K.. Hybrid evolutionary algorithm with marriage of genetic algorithm and extremal optimization for production scheduling. The International Journal of Advanced Manufacturing Technology 2008;36(9-10):959–968.

Eremeev, A.V.. On complexity of optimal recombination for binary representations of solutions. Evolutionary Computation 2008;16(1):127–147.

Glover, F.. Inequalities and target objectives for metaheuristic search–part i: Mixed binary optimization. In: Advances in metaheuristics for hard optimization. Springer; 2008, p. 439–474.

Hu, B., Leitner, M., Raidl, G.R.. Combining variable neighborhood search with integer linear programming for the generalized minimum spanning tree problem. Journal of Heuristics 2008;14(5):473–499.

Hu, B., Raidl, G.R.. Effective neighborhood structures for the generalized traveling salesman problem. In: Evolutionary Computation in Combinatorial Optimization. Springer; 2008, p. 36–47.

Juang, W.S., Su, S.F.. Multiple sequence alignment using modified dynamic programming and particle swarm optimization. Journal of the Chinese institute of engineers 2008;31(4):659–673.

Khichane, M., Albert, P., Solnon, C.. Integration of aco in a constraint programming language. In: Ant Colony Optimization and Swarm Intelligence. Springer; 2008, p. 84–95.

Leitner, M., Raidl, G.R.. Lagrangian decomposition, metaheuristics, and hybrid approaches for the design of the last mile in fiber optic networks. In: Hybrid Metaheuristics. Springer; 2008, p. 158–174.

Maniezzo, V., Roffilli, M.. Very strongly constrained problems: an ant colony optimization approach. Cybernetics and Systems: An International Journal 2008;39(4):395–424.

Meyer, B.. Hybrids of constructive metaheuristics and constraint programming: A case study with aco. In: Hybrid Metaheuristics. Springer; 2008, p. 151–183.

Pirkwieser, S., Raidl, G.R.. Finding consensus trees by evolutionary, variable neighborhood search, and hybrid algorithms. In: Proceedings of the 10th annual conference on Genetic and evolutionary computation. ACM; 2008, p. 323–330.

Prandtstetter, M., Raidl, G.R.. An integer linear programming approach and a hybrid variable neighborhood search for the car sequencing problem. European Journal of Operational Research 2008;191(3):1004–1022.

Puchinger, J., Raidl, G.R.. Bringing order into the neighborhoods: Relaxation guided variable neighborhood search. Journal of Heuristics 2008;14(5):457–472.

Watson, J.P., Beck, J.C.. A hybrid constraint programming/local search approach to the job-shop scheduling problem. In: Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. Springer; 2008, p. 263–277.

Bautista, J., Pereira, J.. A dynamic programming based heuristic for the assembly line balancing problem. European Journal of Operational Research 2009;194(3):787–794.

Birattari, M.. Tuning metaheuristics: a machine learning perspective; vol. 197. Springer; 2009.

Blum, C., Blesa, M.. Solving the kct problem: large-scale neighborhood search and solution merging. Optimization Techniques for Solving Complex Problems 2009;76(22):407–421.

Boschetti, M., Maniezzo, V.. Benders decomposition, lagrangean relaxation and metaheuristic design. Journal of Heuristics 2009;15(3):283–312.

Caserta, M., Voß, S.. A corridor method-based algorithm for the pre-marshalling problem. In: Applications of Evolutionary Computing. Springer; 2009, p. 788–797.

Hentenryck, P.V., Michel, L.. Constraint-based local search. The MIT Press; 2009.

Leitner, M., Raidl, G.R.. Combining lagrangian decomposition with very large scale neighborhoood search for capacitated connected facility location. In: Post-Conference Book of the Eight Metaheuristics International Conference–MIC. 2009,.

Lombardi, M., Milano, M., Roli, A., Zanarini, A.. Deriving information from sampling and diving. In: AI* IA 2009: Emergent Perspectives in Artificial Intelligence. Springer; 2009, p. 82–91.

Pirkwieser, S., Raidl, G.R.. Boosting a variable neighborhood search for the periodic vehicle routing problem with time windows by ilp techniques. In: Proceedings of the 8th metaheuristic international conference (MIC 2009), Hamburg, Germany. 2009a, p. 33–35.

Pirkwieser, S., Raidl, G.R.. Multiple variable neighborhood search enriched with ilp techniques for the periodic vehicle routing problem with time windows. In: Hybrid Metaheuristics. Springer; 2009b, p. 45–59.

Prandtstetter, M.. Hybrid optimization methods for warehouse logistics and the reconstruction of destroyed paper documents. Ph.D. thesis; Ph. D. thesis, Vienna University of Technology; 2009.

Prandtstetter, M., Raidl, G.R.. Meta-heuristics for reconstructing cross cut shredded text documents. In: Proceedings of the 11th Annual conference on Genetic and evolutionary computation. ACM; 2009, p. 349–356.

Prandtstetter, M., Raidl, G.R., Misar, T.. A hybrid algorithm for computing tours in a spare parts warehouse. In: Evolutionary Computation in Combinatorial Optimization. Springer; 2009, p. 25–36.

Prestwich, S., Tarim, S.A., Rossi, R., Hnich, B.. Evolving parameterised policies for stochastic constraint programming. In: Principles and Practice of Constraint Programming-CP 2009. Springer; 2009, p. 684–691.

Rei, W., Cordeau, J.F., Gendreau, M., Soriano, P.. Accelerating benders decomposition by local branching. INFORMS Journal on Computing 2009;21(2):333–345.

Ritzinger, U., Prandtstetter, M., Raidl, G.R.. Computing optimized stock (re-) placements in last-in, first-out warehouses. In: Logistik Management. Springer; 2009, p. 279–298.

Walla, J., Ruthmair, M., Raidl, G.R.. Solving a video-server load re-balancing problem by mixed integer programming and hybrid variable neighborhood search. In: Hybrid Metaheuristics. Springer; 2009, p. 84–99.

Wilbaut, C., Hanafi, S.. New convergent heuristics for 0–1 mixed integer programming. European Journal of Operational Research 2009;195(1):62–74.

Al-Shihabi, S., Ólafsson, S.. A hybrid of nested partition, binary ant system, and linear programming for the multidimensional knapsack problem. Computers & Operations Research 2010;37(2):247–255.

Boschetti, M., Maniezzo, V., Roffilli, M.. Decomposition techniques as metaheuristic frameworks. In: Matheuristics. Springer; 2010, p. 135–158.

Caserta, M., Voß, S.. A math-heuristic algorithm for the dna sequencing problem. In: Learning and Intelligent Optimization. Springer; 2010, p. 25–36.

Gruber, M., Raidl, G.R.. (meta-) heuristic separation of jump cuts in a branch&cut approach for the bounded diameter minimum spanning tree problem. In: Matheuristics. Springer; 2010, p. 209–229.

Khichane, M., Albert, P., Solnon, C.. Strong combination of ant colony optimization with constraint programming optimization. In: Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. Springer; 2010, p. 232–245.

Leitner, M., Leitner, M.. Solving two network design problems by mixed integer programming and hybrid optimization methods. Ph.D. thesis; PhD thesis, Vienna University of Technology, Institute of Computer Graphics and Algorithms, Vienna, Austria (May 2010); 2010.

López-Ibáñez, M., Blum, C.. Beam-aco for the travelling salesman problem with time windows. Computers & operations research 2010;37(9):1570–1583.

Lozano, M., García-Martínez, C.. Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report. Computers & Operations Research 2010;37(3):481–497.

Molina, D., Lozano, M., García-Martínez, C., Herrera, F.. Memetic algorithms for continuous optimisation based on local search chains. Evolutionary Computation 2010;18(1):27–63.

Montemanni, R., Smith, D.H.. Heuristic manipulation, tabu search and frequency assignment. Computers & Operations Research 2010;37(3):543–551.

de Paula, M.R., Mateus, G.R., Ravetti, M.G.. A non-delayed relax-and-cut algorithm for scheduling problems with parallel machines, due dates and sequence-dependent setup times. Computers & Operations Research 2010;37(5):938–949.

Pimmer, M.. A timeslot-based heuristic approach to construct high-school timetables. Ph.D. thesis; Master's thesis, Vienna University of Technology; 2010.

Pirkwieser, S., Raidl, G.R.. Variable neighborhood search coupled with ilp-based very large neighborhood searches for the (periodic) location-routing problem. In: Hybrid Metaheuristics. Springer; 2010a, p. 174–189.

Pirkwieser, S., Raidl, G.R.. Multilevel variable neighborhood search for periodic routing problems. In: Evolutionary Computation in Combinatorial Optimization. Springer; 2010b, p. 226–238.

Pirkwieser, S., Raidl, G.R.. Matheuristics for the periodic vehicle routing problem with time windows. Proceedings of matheuristics 2010c;:28–30.

Puchinger, J., Raidl, G.R., Pirkwieser, S.. Metaboosting: enhancing integer programming techniques by metaheuristics. In: Matheuristics. Springer; 2010a, p. 71–102.

Puchinger, J., Raidl, G.R., Pferschy, U.. The multidimensional knapsack problem: Structure and algorithms. INFORMS Journal on Computing 2010b;22(2):250–265.

Resende, M.G., Martí, R., Gallego, M., Duarte, A.. Grasp and path relinking for the max–min diversity problem. Computers & Operations Research 2010;37(3):498–508.

Schauer, C., Prandtstetter, M., Raidl, G.R.. A memetic algorithm for reconstructing cross-cut shredded text documents. In: Hybrid Metaheuristics. Springer; 2010, p. 103–117.

Applegate, D.L., Bixby, R.E., Chvatal, V., Cook, W.J.. The traveling salesman problem: a computational study. Princeton University Press; 2011.

Berlakovich, M., Ruthmair, M., Raidl, G.R.. A multilevel heuristic for the rooted delay-constrained minimum spanning tree problem. In: Computer Aided Systems Theory–EUROCAST 2011. Springer; 2011, p. 256–263.

Caserta, M., Voß, S., Sniedovich, M.. Applying the corridor method to a blocks relocation problem. OR spectrum 2011;33(4):915–929.

Hanafi, S., Wilbaut, C.. Improved convergent heuristics for the 0-1 multidimensional knapsack problem. Annals of Operations Research 2011;183(1):125–142.

Hu, B., Raidl, G.R.. An evolutionary algorithm with solution archive for the generalized minimum spanning tree problem. In: Computer Aided Systems Theory–EUROCAST 2011. Springer; 2011, p. 287–294.

Jamili, A., Shafia, M.A., Tavakkoli-Moghaddam, R.. A hybrid algorithm based on particle swarm optimization and simulated annealing for a periodic job shop scheduling problem. The International Journal of Advanced Manufacturing Technology 2011;54(1-4):309–322.

Krenek, T.. Verbrauchsminimierung eines hybridfahrzeuges im neuen europäischen fahrzyklus. Master's thesis; Vienna University of Technology, Institute of Computer Graphics and Algorithms, Vienna, Austria; 2011.

Pimmer, M., Raidl, G.R.. A timeslot-filling based heuristic approach to construct high-school timetables. In: Proceedings of the 9th Metaheuristic International Conference (MIC 2011), Udine, Italy. Springer; 2011, p. 349–358.

Pirkwieser, S., Raidl, G.R., Gottlieb, J.. Improved packing and routing of vehicles with compartments. In: Computer Aided Systems Theory–EUROCAST 2011. Springer; 2011a, p. 392–399.

Pirkwieser, S., Raidl, G.R., Gottlieb, J.. Tackling the loading aspect of the vehicle routing problem with compartments. In: Proceedings of the 9th Metaheuristic International Conference (MIC 2011), Udine, Italy. 2011b, p. 25–28.

Archetti, C., Guastaroba, G., Speranza, M.. An ilp-refined tabu search for the directed profitable rural postman problem. Discrete Applied Mathematics 2012;(0):–.

Bortfeldt, A.. A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints. Computers & Operations Research 2012;39(9):2248–2257.

Hiermann, G.. Metaheuristics for a multimodal home-health care scheduling problem. Ph.D. thesis; Vienna University of Technology, Institute of Computer Graphics and Algorithms; 2012.

Jamili, A., Shafia, M., Sadjadi, S., Tavakkoli-Moghaddam, R.. Solving a periodic single-track train timetabling problem by an efficient hybrid algorithm. Engineering Applications of Artificial Intelligence 2012;25(4):793 – 800.

Krenek, T., Ruthmair, M., Raidl, G.R., Planer, M.. Applying (hybrid) metaheuristics to fuel consumption optimization of hybrid electric vehicles. In: Applications of Evolutionary Computation. Springer; 2012, p. 376–385.

Niknam, T., Narimani, M.R., Azizipanah-Abarghooee, R.. A new hybrid algorithm for optimal power flow considering prohibited zones and valve point effect. Energy Conversion and Management 2012;58:197–206.

Pirkwieser, D.S.. Hybrid metaheuristics and matheuristics for problems in bioinformatics and transportation. Ph.D. thesis; Vienna University of Technology; 2012.

Ravetti, M., Riveros, C., Mendes, A., Resende, M., Pardalos, P.. Parallel hybrid heuristics for the permutation flow shop problem. Annals of Operations Research 2012;199(1):269–284.

Sioud, A., Gravel, M., Gagné, C.. A hybrid genetic algorithm for the single machine scheduling problem with sequence-dependent setup times. Computers & Operations Research 2012;39(10):2415–2424.

Subramanian, A., Penna, P.H.V., Uchoa, E., Ochi, L.S.. A hybrid algorithm for the heterogeneous fleet vehicle routing problem. European Journal of Operational Research 2012;221(2):285–295.

Inführ, J., Raidl, G.R.. Grasp and variable neighborhood search for the virtual network mapping problem. In: Hybrid Metaheuristics. Springer; 2013a, p. 159–173.

Inführ, J., Raidl, G.R.. A memetic algorithm for the virtual network mapping problem. In: The 10th Metaheuristics Int. Conference, MIC13, Singapore. Springer; 2013b,.

Inführ, J., Raidl, G.R.. Solving the virtual network mapping problem with construction heuristics, local search and variable neighborhood descent. In: Evolutionary Computation in Combinatorial Optimization. Springer; 2013c, p. 250–261.

Leung, S.C., Zhang, Z., Zhang, D., Hua, X., Lim, M.K.. A metaheuristic algorithm for heterogeneous fleet vehicle routing problems with two-dimensional loading constraints. European Journal of Operational Research 2013;225(2):199–210.

Narimani, M.R., Azizipanah-Abarghooee, R., Zoghdar-Moghadam-Shahrekohne, B., Gholami, K.. A novel approach to multi-objective optimal power flow by a new hybrid optimization algorithm considering generator constraints and multi-fuel type. Energy 2013;49:119–136.

Raidl, G.R., Hu, B., Rainer-Harbach, M., Papazek, P.. Balancing bicycle sharing systems: Improving a vns by efficiently determining optimal loading operations. In: Hybrid Metaheuristics. Springer; 2013, p. 130–143.

Rainer-Harbach, M., Papazek, P., Hu, B., Raidl, G.R.. Balancing bicycle sharing systems: A variable neighborhood search approach. In: Evolutionary Computation in Combinatorial Optimization. Springer; 2013a, p. 121–132.

Rainer-Harbach, M., Papazek, P., Hu, B., Raidl, G.R.. Pilot, grasp, and vns approaches for the static balancing of bicycle sharing systems. In: Technical Report TR 186-1-13-01. Institute of Computer Graphics and Algorithms, Vienna University of Technology; 2013b,.

Ruan, Q., Zhang, Z., Miao, L., Shen, H.. A hybrid approach for the vehicle routing problem with three-dimensional loading constraints. Computers & Operations Research 2013;40(6):1579–1589.

Schöbel, A., Raidl, G., Grujicic, I., Besau, G., Schuster, G.. An optimization model for integrated timetable based design of railway infrastructure. In: In Proceedings of the 5th International Seminar on Railway Operations Modelling and Analysis - RailCopenhagen. IAROR; 2013,.

Solnon, C.. Ant colony optimization and constraint programming. Wiley. com; 2013.

Subramanian, A., Uchoa, E., Ochi, L.S.. A hybrid algorithm for a class of vehicle routing problems. Computers & Operations Research 2013;40(10):2519–2531.

Nogueira, T.H., de Carvalho, C.R.V., Ravetti, M.G.. Analysis of mixed integer programming formulations for single machine scheduling problems with sequence dependent setup times and release dates. 4OR: A Quarterly Journal of Operations Research (submitted) 2014;.

Sousa, J.P., Wolsey, L.A.. A time indexed formulation of non-preemptive single machine scheduling problems. Mathematical Programming 1992;54:353–367.

Pessoa, A., Uchoa, E., Aragão, M., Rodrigues, R.. Exact algorithm over an arc-time-indexed formulation for parallel machine scheduling problems. Mathematical Programming Computation 2010;2:259–290.

Fisher, M.L.. Optimal solution of scheduling problems using lagrange multipliers: Part i. Operations Research 1973;21:1114–1127.

Geoffrion, A.M.. Lagrangian relaxation applied to capacitated facility location problems. AIIE Transation 1978;10:40–47.

Minoux, M.. Theory and Algorithms. Mathematical Programming, Wiley; 1986.

Lasdon, L.S.. Optimization Theory for Large Systems. Dover Publications, INC; 2002.

Guignard, M.. Lagrangean relaxation. TOP 2003;11(2):151–228.

Lucena, A.. Non delayed relax-and-cut algorithms. In: In IFIP 74 Congress. 1974, p. 552–556.

de Paula, M.R., Ravetti, M.G., Mateus, G.R.. Solving parallel machines scheduling problems with sequence-dependent setup times using variable neighbourhood search. IMA Journal of Management Mathematics 2007;18:101–115.

Barahona, F., Anbil, R.. The volume algorithm: producing primal solutions with a subgradient method. Mathematical Programming 2000;87(3):385–399.

Barahona, F., Ladányi, L.. Branch and cut based on the volume algorithm: Steiner trees in graphs and max-cut. RAIRO - Operations Research 2006;40:53–73.

Fukuda, E.H.. Algoritmo de volume e otimização não diferenciável. Ph.D. thesis; Dissertação de Mestrado - USP; 2007.

Lemaréchal, C.. An extension of davidon methods to non differentiable problems. In: Nondifferentiable optimization. Springer; 1975, p. 95–109.

Wolfe, P.. A method of conjugate subgradients for minimizing nondifferentiable functions. In: Nondifferentiable optimization. Springer; 1975, p. 145–173.

Lemaréchal, C.. Nondifferentiable optimization. In: In: Optimization, Hanbooks in Operations Research. Nemhauser, G.L., Rinnooy Kan, A.H.G., Todd, M.J. (eds.). North Holland; 1989, p. 529–572.

Lemaréchal, C.. Lagrangian relaxation, Computational Combinatorial. Springer Verlag; 2001.

Bahiense, L., Maculan, N., Sagastizabal, C.A.. The volume algorithm revised: relation with bundle methods. Mathematical Programming 2002;94:41–69.

Nawaz, M., Enscore Jr, E.E., Ham, I.. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. Omega 1983;11(1):91–95.

Naderi, B., Ruiz, R., Zandieh, M.. Algorithms for a realistic variant of flowshop scheduling. Computers & Operations Research 2010;37(2):236–246.

Stutzle, T.. Local search algorithms for combinatorial problems. Ph.D. thesis; PhD Thesis - Darmstadt University of Technology; 1998.

Stutzle, T., Hoos, H.H., Ribeiro, C.C., Hansen, P.. Analyzing the runtime behaviour of iterated local search for the travelling salesman problem. In: Essays and surveys on metaheuristics. Kluwer Academic publishers; 2001, p. 589–611.

Hoos, H.H., Stützle, T.. Stochastic local search: Foundations & applications. Morgan Kaufmann; 2004.

Baxter, J.. Local optima avoidance in depot location. Journal of the Operational Research Society 1981;32:815–819.

Baum, E.B.. Towards practical 'neural' computation for combinatorial optimization problems. In: AIP Conference Proceedings; vol. 151. 1986a, p. 53.

Baum, E.B.. Iterated descent: A better algorithm for local search in combinatorial optimization problems. Tech. Rep.; Caltech, Pesadena, CA; 1986b.

Johnson, D.. Local optimization and the traveling salesman problem. In: Paterson, M., editor. Automata, Languages and Programming; vol. 443 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg; 1990, p. 446–461.

Martin, O., Otto, S.W., Felten, E.W.. Large-step Markov chains for the traveling salesman problem. Citeseer; 1991.

Martin, O.C., Otto, S.W.. Combining simulated annealing with local search heuristics. Annals of Operations Research 1996;63(1):57–75.

Brucker, P., Hurink, J., Werner, F.. Improving local search heuristics for some scheduling problems- part i. Discrete Applied Mathematics 1996;65(1):97–122.

Brucker, P., Hurink, J., Werner, F.. Improving local search heuristics for some scheduling problems - part ii. Discrete Applied Mathematics 1997;72(1):47–69.

Johnson, D.S., McGeoch, L.A.. The traveling salesman problem: A case study in local optimization. John Wiley, New York; 1997.

Applegate, D., Cook, W., Rohe, A.. Chained lin-kernighan for large traveling salesman problems. INFORMS Journal on Computing 2003;15(1):82–92.

Rodrigues, R., Pessoa, A., Uchoa, E., Poggi de Aragão, M.. Heuristic algorithm for the parallel machine total weighted tardiness scheduling problem. Tech. Rep.; internal report 10/2008, Universidad Federal Fluminense; 2008.

Lourenço, H.R., Martin, O.C., Stützle, T.. Iterated local search: Framework and applications. In: Gendreau, M., Potvin, J.Y., editors. Handbook of Metaheuristics; vol. 146 of *International Series in Operations Research & Management Science*. Springer US. ISBN 978-1-4419-1663-1; 2010, p. 363–397.

Arroyo, J.E.C., Nunes, G.V.P., Kamke, E.H.. Iterative local search heuristic for the single machine scheduling problem with sequence dependent setup times and due dates. In: Hybrid Intelligent Systems, 2009. HIS'09. Ninth International Conference on; vol. 1. IEEE; 2009, p. 505–510.

Den Besten, M., Stützle, T., Dorigo, M.. Design of iterated local search algorithms: An example application to the single machine total weighted tardiness problem. In: Evo Workshop: LNCS 2037, pp. 441-451. 2001,.

Congram, R.K., Potts, C.N., Van De Velde, S.L.. An iterated dynasearch algorithm for the single-machine total weighted tardiness scheduling problem. INFORMS Journal on Computing 2002;14(1):52–67.

Rubin, P.A., Ragatz, G.L.. Scheduling in a sequence dependent setup environment with genetic search. Computers & Operations Research 1995;22(1):85–99.

Graham, R., Lawler, E., Lenstra, J., Kan, A.. Optimization and approximation in deterministic sequencing and scheduling: a survey. In: P.L. Hammer, E.J., Korte, B., editors. Discrete Optimization II Proceedings of the Advanced Research Institute on Discrete Optimization and Systems Applications of the Systems Science Panel of NATO and of the Discrete Optimization Symposium co-sponsored by IBM Canada and SIAM Banff, Aha. and Vancouver; vol. 5 of *Annals of Discrete Mathematics*. Elsevier; 1979, p. 287 – 326.

Gupta, S.R., Smith, J.S.. Algorithms for single machine total tardiness scheduling with sequence dependent setups. European Journal of Operational Research 2006;175(2):722–739.

Liao, C.J., Juan, H.C.. An ant colony optimization for single-machine tardiness scheduling with sequence-dependent setups. Computers & Operations Research 2007;34(7):1899–1909.

Rardin, R.L., Uzsoy, R.. Experimental evaluation of heuristic optimization algorithms: A tutorial. Journal of Heuristics 2001;7(3):261–304.