Lecture Notes in Artificial Intelligence 6404

Edited by R. Goebel, J. Siekmann, and W. Wahlster

Subseries of Lecture Notes in Computer Science

Antônio Carlos da Rocha Costa
Rosa Maria Vicari
Flavio Tonidandel (Eds.)

Advances in Artificial Intelligence – SBIA 2010

20th Brazilian Symposium on Artificial Intelligence
São Bernardo do Campo, Brazil, October 23-28, 2010
Proceedings

Springer

Volume Editors

Antônio Carlos da Rocha Costa
Universidade Federal do Rio Grande – FURG
Centro de Ciências Computacionais
Av. Itália, km 8 – Campus Carreiros, 96.201-900 Rio Grande, RS, Brazil
E-mail: ac.rocha.costa@gmail.com

Rosa Maria Vicari
Universidade Federal do Rio Grande do Sul – UFRGS
Instituto de Informática
Av. Bento Gonçalves 9.500, 91501-970 Porto Alegre, RS, Brazil
E-mail: rosa@inf.ufrgs.br

Flavio Tonidandel
Centro Universitário da FEI
Departamento de Ciência da Computação
Av. Humberto A. C. Branco 3972, 09850-901 São Bernardo do Campo, SP, Brazil
E-mail: flaviot@fei.edu.br

Library of Congress Control Number: 2010935023

CR Subject Classification (1998): I.2, H.3, H.4, I.4, I.5, H.5

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-642-16137-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-16137-7 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

The SBIA conference series started in 1984 at the Federal University of Rio Grande do Sul (UFRGS) and through the years has benefited the Artificial Intelligence and Computer Science communities in Brazil.

After 26 years and 20 conferences SBIA is now a mature event, constituting a discussion forum for new ideas in all sub-areas of AI.

In this book you will find the full papers selected for publication in the SBIA 2010 proceedings. The papers cover the AI sub-areas in the following way:

– Ontologies, Knowledge Representation, and Reasoning: 8
– Machine Learning: 2
– Autonomous Agents and Multiagent Systems: 6
– Natural Language Processing: 2
– Planning and Scheduling: 5
– Logics for AI: 3
– Constraints and Search: 5

We would like to thank all the authors that contributed to SBIA 2010. We also thank all the members of the international Program Committee and the additional reviewers, who did an excellent job in reviewing the papers.

We are very grateful to Flavio Tonidandel, General Chair of SBIA 2010 and of the Joint SBIA/SBRN/JRI 2010 Conference, for all the support that he and his team at FEI provided.

Yoav Shoham, Jaime Sichman, and David Hogg were the keynote speakers of the event. We thank them very much for their acceptance of the invitation.

A special acknowledgement is due to Tiago Thompsen Primo, for his dedicated effort in the editing of these proceedings.

Finally, we thank the SBIA 2010 sponsors (CAPES, CNPq, FAPESP, and SBC) for their support.

August 2010

Antônio Carlos da Rocha Costa
Rosa Maria Vicari

Organization

Organizing Committee

General Chair

Flavio Tonidandel	Centro Universitário da FEI

Workshop Chair

Plínio Thomaz Aquino Jr.	Centro Universitário da FEI

Tutorial Chair

Paulo Sérgio Silva Rodrigues	Centro Universitário da FEI

Steering Committee (CEIA)

Augusto Loureiro da Costa	Universidade Federal da Bahia (Coordinator)
Solange Oliveira Rezende	Universidade de São Paulo
Fernando Santos Osório	Universidade de São Paulo
Fabio Gagliardi Cozman	Universidade de São Paulo
Marcelo Finger	Universidade de São Paulo
Fred Freitas	Universidade Federal de Pernambuco
Alexandre da Silva Simões	Universidade Estadual Paulista

Program Committee

Adina Magda Florea	Polytechnic University of Bucharest, Romania
Adolfo Arenas	Inst. Politec. Nal. Campus "Lopez Mateos", Mexico
Adolfo Neto	UTFPR, Brazil
Adriano Werhli	FURG, Brazil
Alejandro Zunino	ISISTAN-UNICEN, Argentina
Alessio Lomuscio	Imperial College, UK
Alexander Gelbukh	National Polytechnic Institute, Mexico
Alexandre Silva	UFRJ, Brazil
Aline Villavicencio	UFRGS, Brazil
Alneu Lopes	USP, Brazil
Alvaro Moreira	UFRGS, Brazil
Amilcar Cardoso	University of Coimbra, Portugal
Ana Bazzan	UFRGS, Brazil
Ana Bicharra	UFF, Brazil

Ana Casali	Universidad Nacional de Rosario, Argentina
Ana Carolina Bertoletti De Marchi	UPF, Brazil
Ana Teresa Martins	UFC, Brazil
Anarosa Brandao	USP, Brazil
André Campos	UFRN, Brazil
André Ponce de Leon F. de Carvalho	USP, Brazil
Andrea Omicini	University of Bologna, Italy
Anna Helena Reali Costa	USP, Brazil
Anne Canuto	UFRN, Brazil
Antônio Braga	UFMG, Brazil
Antonio Carlos da Rocha Costa	FURG, Brazil
Artur Garcez	City University of London, UK
Augusto Loureiro da Costa	UFBA, Brazil
Aurora Pozo	UFPR, Brazil
Barbara Hammer	Clausthal University of Technology, Germany
Benjamin Bedregal	UFRN, Brazil
Bianca Zadrozny	UFF, Brazil
Blai Bonet	Universidad Simón Bolívar, Venezuela
Carlos Reyes-Garcia	INAOE, Mexico
Carlos Ribeiro	ITA, Brazil
Cassio de Campos	Dalle Molle Institute for AI, Switzerland
Celso Kaestner	UTFPR, Brazil
Christian Lemaitre	Universidad Autónoma Metropolitana, Mexico
Clarisse de Souza	PUC-Rio, Brazil
Daniel Berrar	University of Ulster, UK
Diana Adamatti	FURG, Brazil
Donato Malerba	Università degli Studi di Bari, Italy
Edson Matsubara	UFMS, Brazil
Edward Hermann Haeusler	PUC-Rio, Brazil
Eric Matson	Purdue University, USA
Eugenio Oliveira	Universidade do Porto, Portugal
Evandro Costa	UFAL, Brazil
Fabiano Silva	UFPR, Brazil
Fariba Sadri	Imperial College, UK
Fábio Cozman	USP, Brazil
Federico Barber	Universidad Politecnica de Valencia, Spain
Fernando Osório	USP, Brazil
Filip Zelezny	Czech Techn. University in Prague, Czech Republic
Flavio Tonidandel	FEI, Brazil
Flávio Soares Corra da Silva	USP, Brazil
Francisco de Carvalho	UFPE, Brazil

Frank Dignum	Utrecht University, Netherlands
Fred Freitas	UFPE, Brazil
Gabriel Lopes	Universidade Nova de Lisboa, Portugal
Geber Ramalho	UFPE, Brazil
Gerardo Schneider	University of Gothenburg, Sweden
Gerson Zaverucha	UFRJ, Brazil
Graça Gaspar	Universidade de Lisboa, Portugal
Graçaliz Dimuro	FURG, Brazil
Guilherme Barreto	UFC, Brazil
Guillermo Simari	Universidad Nacional del Sur, Argentina
Gustavo Giménez-Lugo	UTFPR, Brazil
Hector Geffner	Universitat Pompeu Fabra, Spain
Helder Coelho	Universidade de Lisboa, Portugal
Heloisa Camargo	UFSCar, Brazil
Ines Dutra	Universidade do Porto, Portugal
Irwin King	The Chinese University of Hong Kong, China
Ivandré Paraboni	USP, Brazil
Jaime Sichman	USP, Brazil
James Cussens	University of York, UK
Jan Ramon	Katholieke Universiteit Leuven, Belgium
João Gama	Universidade do Porto, Portugal
João Marcos	UFRN, Brazil
João Balsa	Universidade de Lisboa, Portugal
João Luis Tavares da Silva	UCS, Brazil
Jomi Hübner	UFSC, Brazil
Juergen Dix	Clausthal University of Technology, Germany
Julie Dugdale	Laboratoire d'Informatique de Grenoble, France
Krysia Broda	Imperial College, UK
Laurent Perrussel	Université de Toulouse, France
Lúcia Drummond	UFF, Brazil
Leliane Nunes de Barros	USP, Brazil
Leonardo Emmendorfer	FURG, Brazil
Li Weigang	UnB, Brazil
Luis Antunes	Universidade de Lisboa, Portugal
Luis Fariñas del Cerro	Université Paul Sabatier, France
Luis Lamb	UFRGS, Brazil
Luis Otávio Alvares	UFRGS, Brazil
Luiz Satoru Ochi	UFF, Brazil
Luiza Mourelle	UERJ, Brazil
Mara Abel	UFRGS, Brazil
Marcelo Finger	USP, Brazil
Marcilio de Souto	UFRN, Brazil
Marco Gori	University of Siena, Italy

Toby Walsh	NICTA and Univ. New South Wales, Australia
Torsten Schaub	University of Potsdam, Germany
Virginia Dignum	Delft Univ. of Technology, Netherlands
Valerie Camps	Université Paul Sabatier, France
Vania Bogorny	UFSC, Brazil
Vítor Santos Costa	Universidade do Porto, Portugal
Vera Lúcia Strube de Lima	PUCRS, Brazil
Viviane Torres da Silva	UFF, Brazil
Vladik Kreinovich	University of Texas at El Paso, USA
Wagner Meira, Jr.	UFMG, Brazil
Walter Carnielli	UNICAMP, Brazil
Walter Daelemans	University of Antwerp, Belgium
Wamberto Vasconcelos	University of Aberdeen, UK
Wiebe van der Hoek	University of Liverpool, UK
Yves Demazeau	Laboratoire d'Informatique de Grenoble, France
Zhao Liang	USP, Brazil

Additional Reviewers

André Lins Aquino	UFOP, Brazil
Caecilia Zirn	Univ. Heidelberg, Germany
Cassia Santos	Evora University, Portugal
Christian Vogler	ILSP, Greece
Daniel Weingaertner	Universidade Federal do Paraná, Brazil
Eder Mateus Gonçalves	FURG, Brazil
Eugenio Silva	PUC-Rio, Brazil
Francicleber Ferreira	UFC, Brazil
Frédéric Moisan	Université de Toulouse, France
Gustavo Batista	USP, Brazil
Illya Kokshenev	UFMG, Brazil
João Alcântara	UFC, Brazil
Karla Figueiredo	PUC-Rio, Brazil
Luis Künzle	UFPR, Brazil
Martin Musicante	UFRN, Brazil
Natasha Lino	UFPb, Brazil
Nicolau Werneck	USP, Brazil
Paulo Schreiner	UFRGS, Brazil
Regivan Hugo Nunes Santiago	UFRN, Brazil
Rinaldo Lima	UFPE, Brazil
Rodrigo Wilkens	UFRGS, Brazil
Rosiane de Freitas Rodrigues	UFAM, Brazil
Sérgio Almeida	UCPel, Brazil
Thiago Noronha	UFMG, Brazil

Supporting Scientific Society

SBC - Sociedade Brasileira de Computação.

Sponsoring Institutions

CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológica
CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
FAPESP - Fundação de Amparo à Pesquisa do Estado de São Paulo

Table of Contents

Chapter 1: Ontologies, Knowledge Representation and Reasoning

Ontological Primitives for Visual Knowledge 1
Alexandre Lorenzatti, Mara Abel, Sandro Rama Fiorini, Ariane Kravczyk Bernardes, and Claiton Marion dos Santos Scherer

A Semi-automatic Method for Domain Ontology Extraction from Portuguese Language Wikipedia's Categories 11
Clarissa Castellã Xavier and Vera Lucia Strube de Lima

Ontology Reasoning in Agent-Oriented Programming 21
Claudio Fuzitaki, Álvaro Moreira, and Renata Vieira

System Design Modification with Actions 31
Maria Viviane de Menezes, Silvio do Lago Pereira, and Leliane Nunes de Barros

Learning Terminologies in Probabilistic Description Logics 41
Kate Revoredo, José Eduardo Ochoa-Luna, and Fabio Gagliardi Cozman

Knowledge-Based System for the Maintenance Registration and Consistency among UML Diagrams 51
Cleverton Ferreira Borba and Ana Estela Antunes da Silva

Semantic Mapping with a Probabilistic Description Logic 62
Rodrigo Polastro, Fabiano Corrêa, Fabio Cozman, and Jun Okamoto Jr.

Markov Decision Processes from Colored Petri Nets 72
Monica Góes Eboli and Fabio Gagliardi Cozman

Chapter 2: Machine Learning

Incremental Learning of Multivariate Gaussian Mixture Models 82
Paulo Martins Engel and Milton Roberto Heinen

Bayesian Network Structure Inference with an Hierarchical Bayesian Model 92
Adriano Velasque Werhli

Chapter 3: Autonomous Agents and Multiagent Systems

On the Construction of Synthetic Characters with Personality and Emotion 102
Ary Fagundes Bressane Neto and Flávio Soares Corrêa da Silva

Towards Automated Trading Based on Fundamentalist and Technical Data 112
Carlos Henrique Dejavite Araújo and Paulo André Lima de Castro

Developing a Consciousness-Based Mind for an Artificial Creature 122
Ricardo Capitanio Martins da Silva and Ricardo Ribeiro Gudwin

Simulating the Emergence of Social Relationship Networks in Groups of Believable Agents: The X-BARIM Model 133
Pablo Barbosa, Danielle Silva, Geber Ramalho, and Patricia Tedesco

Using Jason to Develop Normative Agents 143
Baldoino Fonseca dos Santos Neto, Viviane Torres da Silva, and Carlos José Pereira de Lucena

Improving Space Representation in Multiagent Learning via Tile Coding 153
Samuel Justo Waskow and Ana Lúcia Cetertich Bazzan

Chapter 4: Natural Language Processing

Factored Translation between Brazilian Portuguese and English 163
Helena de Medeiros Caseli and Israel Aono Nunes

Question Answering for Portuguese: How Much Is Needed? 173
Rodrigo Wilkens and Aline Villavicencio

Chapter 5: Planning and Scheduling

Planning for Multi-robot Localization 183
Paulo Pinheiro and Jacques Wainer

Symbolic Bounded Real-Time Dynamic Programming 193
Karina Valdivia Delgado, Cheng Fang, Scott Sanner, and Leliane Nunes de Barros

An Adaptive Genetic Algorithm to the Single Machine Scheduling Problem with Earliness and Tardiness Penalties 203
Fábio Fernandes Ribeiro, Marcone Jamilson Freitas Souza, and Sérgio Ricardo de Souza

A Dijkstra Algorithm for Fixed-Wing UAV Motion Planning Based on Terrain Elevation ... 213
Felipe Leonardo Lôbo Medeiros and José Demisio Simões da Silva

Feasible UAV Path Planning Using Genetic Algorithms and Bézier Curves ... 223
Douglas Guimarães Macharet, Armando Alves Neto, and Mario Fernando Montenegro Campos

Chapter 6: Constraints and Search

High-Level Modeling of Component-Based CSPs ... 233
Raphaël Chenouard, Laurent Granvilliers, and Ricardo Soto

Improving the Distributed Constraint Optimization Using Social Network Analysis ... 243
Allan Rodrigo Leite, André Pinz Borges, Laércio Martins Carpes, and Fabrício Enembreck

A Survey and Classification of A* Based Best-First Heuristic Search Algorithms ... 253
Luis Henrique Oliveira Rios and Luiz Chaimowicz

Chapter 7: Logics for AI

A Sequent Calculus for 3-Dimensional Space ... 263
Norihiro Kamide

Intuitionistic Fuzzy Probability ... 273
Claudilene Gomes Da Costa, Benjamin Callejas Bedregal, and Adrião Duarte Doria Neto

A Proof System for Temporal Reasoning with Sequential Information ... 283
Norihiro Kamide

A Refuted Conjecture on Probabilistic Satisfiability ... 293
Marcelo Finger and Glauber De Bona

A Logic for Conceptual Hierarchies ... 303
Norihiro Kamide

Author Index ... 313

An Adaptive Genetic Algorithm to Total Tardiness Single Machine Scheduling Problem with Earliness and Tardiness Penalties

Fábio Fernandes Ribeiro, Marcone Jamilson Freitas Souza, and Sérgio Ricardo de Souza

Centro Federal de Educação Tecnológica de Minas Gerais - CEFET/MG, Programa de Pos-Graduação em Modelagem Matemática e Computacional,
Avenida Amazonas 7675, Belo Horizonte, CEP: 30510-000, Minas Gerais, Brazil
fabiobh@gmail.com
marcone@iceb.ufop.br
sergio@dppg.cefetmg.br
`http://www.mmc.cefetmg.br`

Abstract. *This paper deals with the Single Machine Scheduling Problem with Earliness and Tardiness Penalties, considering distinct due windows and sequence-dependent setup time. Due to its complexity, an adaptive genetic algorithm is proposed for solving it. Many search operators are used to explore the solution space where the choice probability for each operator depends on the success in a previous search. The initial population is generated by the combination between construct methods based on greedy, random and GRASP techniques. For each job sequence generated, a polynomial time algorithm are used for determining the processing initial optimal date to each job. During the evaluation process, the best individuals produced are add to a special group, called elite group. The individuals of this group are submitted to refinement, looking for improve his quality. Three variations of this algorithm are submitted to computational test. The results shows the effectiveness of the proposed algorithm.*

Key words: Single machine scheduling, Genetic algorithm, metaheuristics.

1 Introduction

Nowadays, scheduling problems are one of the most studied problems [1]. It occurs mainly by two aspects: the first one concerns their practical importance, with various applications in several industries, like chemical, metallurgic and textile industries. The second aspect is about the difficulty for solving the majority problems of this class. This paper deals with the Single Machine Scheduling Problem with Earliness and Tardiness Penalties (SMSPETP) with distinct due windows and sequence-dependent setup time. To our knowledge, this problem has not been still object of great attention of the scientific community, as it could seen in the recent survey [1].

The criteria to penalize the tardiness and earliness production goes to the Just-in-Time philosophy goal where they produce just when necessary. The due window existence for each job, according to [2], is due to an uncertainly situation or tolerance related to due date. We accept that this time interval operations can be finalized without costs. On the other hand, in industrial processes majority, the machines can be prepared to do new jobs, including the time to obtain tools, positioning materials that will be used in the process, cleaning process, preparing process, tools adjustment and materials inspection. The time necessary to this preparation is known by setup time. Many production scheduling researches disregard this time or include it in the operation processing time. This act simplifies the analysis but affect the solution quality when the setup time has a relevant variability in function of the job sequence in machine. This work considers that the setup times are dependents from the production scheduling. Since it was showed in [3] that a simplified version of this problem is NP-Hard, the application of metaheuristics for solving this problem is justified.

To solve this scheduling problem with the characteristics presented, an adaptive genetic algorithm is proposed here, where the initial population was generated by a construction method based on GRASP [5] and five dispatch rules. During the evaluation process, the population passes through mutation and crossover conventional process. However, the crossover uses criteria based on solution quality generated by each crossover operator to choose which operator will be used. By the way, during a determined number of generations, the probability of choice of a operator is update according to the solution quality for each operator. A local search is then applied in the best produced offspring, for each operator, to refine them. The survival population are composed for 95% offspring choose by the elitism procedure. The others 5% are randomly chosen and pass through the mutation process, when replacement between two jobs will be done to warranty the population diversity. Path Relinking will be applied to offspring that pass through local search to allow reconnections between the best offspring generated, like a strategy to find intermediate solutions. The population improvement occurs until the stop criteria will be reached.

This work is organized as follows: section 2 details the studied problem; section 3 presents the adaptive algorithm for solving SMSETP; section 4 shows and discuss the results; finally, section 5 ends this work and shows future perspectives to improve the proposed algorithm.

2 Problem description

This work studies the single machine scheduling problem, with earliness and tardiness penalties, distinct due windows and sequence-dependent setup time. In this problem, one machine must process a set of n jobs. Each job i has processing time P_i, initial date E_i and final date T_i, desired for ending the processing. The machine executes one job per time and, if a job processing is started, it must be finished, since processing interruptions are not allowed. All jobs are available for processing in the instant 0. When a job j is sequenced immediately after a job i, for setting the machine is necessary a setup time S_{ij}. Setup times equal 0 mean products of the same family. The initial setup times are considered, i.e., the setup time to the first job in the sequence is 0. The idle time between the execution with two consecutive jobs is allowed. The jobs must be finalized inside the time interval $[E_i, T_i]$, called due window. In case of job finalization before E_i, there is a cost to earliness. Case the job are finalized after T_i, a cost will be generated for tardiness. For jobs completed within due windows, none cost is incurred. The costs to earliness and tardiness of production depend on jobs. Each job i have a earliness cost α_i and a tardiness cost β_i. Finally, the objective of the problem is to minimize the summation of the earliness and tardiness penalties.

The mixed integer programming model (MIP) below, based on [6], formulates the scheduling problem described above. This formulation uses the following notation:

- s_i: the starting time of job i;
- C_i: the completion time of job i;
- y_{ij}: binary variable that assumes value 1 if job j is processed immediately after job i and 0, otherwise;
- e_i: the earliness of job i, that is, $e_i = \max\{0, E_i - C_i\}$;
- t_i: the tardiness of job i, that is, $t_i = \max\{0, C_i - T_i\}$;
- M: a sufficiently large number;
- 0: a fictitious job, which precedes and succeeds all other jobs;

It also assumes that $P_0 = 0$, $S_{0i} = S_{i0} = 0 \;\; \forall i \in \{1, 2, \cdots, n\}$

$$\min \quad Z = \sum_{i=1}^{n} (\alpha_i e_i + \beta_i t_i) \tag{1}$$

$$\text{s.t:} \quad s_j - s_i - y_{ij}(M + S_{ij}) \geq P_i - M \;\; \forall\, i = 0, \ldots, n; \quad \forall\, j = 0, \ldots, n; \;\; i \neq j \tag{2}$$

$$\sum_{j=0,\; j\neq i}^{n} y_{ij} = 1 \qquad \forall\, i = 0, \ldots, n \tag{3}$$

$$\sum_{i=0,\; i\neq j}^{n} y_{ij} = 1 \qquad \forall\, j = 0, \ldots, n \tag{4}$$

$$s_i + P_i + e_i \geq E_i \qquad \forall\, i = 1, \ldots, n \tag{5}$$

$$s_i + P_i - t_i \leq T_i \qquad \forall\, i = 1, \ldots, n \tag{6}$$

$$s_i \geq 0 \qquad \forall\, i = 0, \ldots, n \tag{7}$$

$$e_i \geq 0 \qquad \forall\, i = 1, \ldots, n \tag{8}$$

$$t_i \geq 0 \qquad \forall\, i = 1, \ldots, n \tag{9}$$

$$y_{ij} \in \{0, 1\} \qquad \forall\, i, j = 0, \ldots, n \tag{10}$$

The objective function (1) expresses the total earliness and tardiness cost. The constraints (2) establish that job j can be processed when job i is finished and the machine is prepared to processes it. The constraints (3), (4) and (10) guarantee that the variable y_{ij} assumes value 1 if and only if job j is processed immediately after job i. The constraints (5) and (6) define, respectively, the tardiness and earliness values according of the due window. The constraints (7) to (10) define the type of the variables.

2.1 Heuristic framework

In this section, the adaptive genetic algorithm framework is described.

- **Individual representation**: An individual (solution) to this problem is represented by a vector v of n genes (jobs), with position i of each gene showing the production sequence of job v_i. For example, in the sequence $v = \{7, 1; 5; 6; 4; 3; 2\}$, the job 7 is the first to be processed and job 2, the last.
- **Evaluation of individuals** All of individuals are evaluated by the same objective function, gives by the expression (1) of MIP model (Mixed Integer Programming), where the individual which obtained the shortest value to objective function are considerer the most adapted.
- **Initial population construction** The initial population of adaptive genetic algorithm proposed is generated by GRASP construction phase ([5]), having as guide function five dispatch rules (EDD, TDD, SPT, WSPT and LPT). For each construction (GRASP + Dispatch rule) 200 individuals are generated. Then individuals are ordered, from the best to the worst, according to evaluated function. Initial population is composed by the best 100 individuals generated.
- **GRASP construction procedure** In this construction procedure, an offspring is formed by genes that are inserted one by one. The offspring is constructed according with a partially greedy selection criteria. To estimate the insertion benefit of each gene, dispatch rules EDD, TDD, SPT, WSPT and LPT is used. Each variant gives a different construction. In Figure 1, the GRASP construction phase is showed. In this figure, E_{min} represents the earliest date to the delivery order and E_{max}, the tardiest date.

2.2 Adaptive Genetic Algorithm applied to SMSETP

Figure 2 shows the pseudo-code of the proposed Adaptive Genetic Algorithm (AGA). The algorithm phases are described in the following.

Individual selection method After population evaluate, the individuals are selected by the tournament method where the mainly goal is allowed that the most adapted individuals are selected.

Crossover After population evaluate, the individuals are selected to reproduction by the selection method already described. The crossover process uses the following operators: (i) One Point Crossover (OX), (ii) Similar Job Order Crossover (SJOX), (iii) Relative Job Order Crossover (RRX), (iv) Based Order Uniform Crossover (BOUX) and (v) Partially Mapped Crossover (PMX). This

```
procedure Construction(g(.), γ, v);
  v ← ∅;
  Initialize a set C of candidate genes;
  while (C ≠ ∅) do
      g(t_min) = min{g(t) | t ∈ C};
      g(t_max) = max{g(t) | t ∈ C};
      RCL = {t ∈ C | g(t) ≤ g(t_min) + γ(g(t_max) − g(t_min))};
      Select, randomly, a gene t ∈ RCL;
      v ← v ∪ {t};
      Update C;
  end-while;
  Return v;
end Construction;
```

Fig. 1. Procedure to build an individual.

```
Algorithm AGA(maxger, nind, probcross, probmut);
  t ← 0;
  Generate Initial Population P(t);
  Evaluate P(t);
  while (t ≤ maxger) do
      t ← t + 1;
      Generate P(t) by P(t − 1);
      while (i ≤ numind) do
          i ← 1;
          cross ← Randomly number from 1 to 100;
          if (cross ≤ probcrossover) then
              Select individual;
              Crossover;
          end-if;
          Evaluate P(t);
      end-while;
      Define survivors;
      if (t mod 5 = 0) then
          Update choose probability of operators (p_(O_i));
          Execute Local Search;
          Apply Path Relinking;
      end-if;
  end-while;
end AGA;
```

Fig. 2. Pseudo-code of the proposed Adaptive Genetic Algorithm

choice was taken by the fact of this operators being the most common operators to solve problems like this by genetic algorithm [4]. The choice probability of crossover operator change according the quality of individuals produced by the operators in the past generations. More specifically, let O_i, with $i = 1, \cdots, 5$, the five *crossover* operators. Initially, each *crossover* operator O_i has the same probability to be choose, that means, $p(O_i) = 1/5$. Being $f(s*)$ the best individual found and A_i the average value individual found for each operator O_i since the last update. Case the operator did not be choose in last five generations, make $A_i = 1$. Being $q_i = f(s^*) = A_i$ and $p(O_i) = q_i / \sum_{j=1}^{5} q_j$ where $i = 1, \cdots, 5$. Observe that how much better the individual are, more are the value of q_i and, consequently, more $p(O_i)$ probability of choose O_i operator. Therefore, during the algorithm evolution, the best operator have its chance of choice increased. This procedure is inspired in *Reactive GRASP* algorithm, proposed by [7].

Local search Like said previously, at each five generations, a local search is applied to the best individual generated by each *crossover* operator. The local search used are Random descending, that use two kinds of movement to explore search space: the change of two jobs of the the sequence and the job relocation to another production sequence. The method works as follow: To an individual, two jobs are selected randomly and the positions are exchanged. If the new movement is better than previously, according to the evaluate function, it is accept and pass to be the current solution; otherwise, another movement are randomly chosen. If during *MRDmax* any solution better than current are generated, so relocate movements are used. If there are improvement, the method return to use exchange movements; otherwise, the local search are ended up when *MRDmax* iterations without improvement happens.

Path Relinking During the evolutive process, a group with the best five individuals generated by each *crossover* operator are build. So, at five generations, Path Relinking are triggered taking as base solution the best individual generated by the method and as guide individual each one of the five best individuals generated by each crossover operator. This procedure is called Truncated Backward Path Relinking, and when 75% of guide individual added to the base solution, procedure are stopped. It considered like attribute a job position of production sequence. For each job candidate to insertion, a local search method are applied like described previously, and a movement of a candidate job are not allowed.

Individual survival The individuals survivors are certain by the elitism technique. The 95% of the most adapted individuals will survive and the others 5% are composed by individuals randomly chosen of current population and submitted to mutation, when the production sequence of two jobs are exchanged.

Stop criteria The maximum number of generations is used to a stop criteria of adaptive evolutionary algorithm.

Table 1. Adaptive Genetic Algorithm parameters

Parameters	Values
Parameter γ GRASP construction phase	0.20
Maximum iterations of local search ($MRDmax$)	$7 \times n$
Maximum generations of AGA ($maxger$)	100
Crossover probability	80%
Mutation rate	5%

Variants of proposed algorithm In this work, three variants of AGA are development. In variant 1, called AGA1, the refresh of crossover operator selection rate happens at each five generations. After that all elite group members are submitted to local search (see local search section) and then to path relinking. In this variant, the elite group are composed by the best individual produced by each crossover operator in last five generations. Variant 2, called AGA2, differs to AGA1 variant by elite group composition. In variant 2, elite group are composed by the best individual produced by each crossover operators globally and not just at last five generations. In Variant 3, called AGA3, the the refresh of crossover operator selection rate, elite group submission to local search and path relinking happens at each ten generations. The elite group are composed by the best three individuals produced globally, by the best solution produced at past ten generation, if this individual have diversity index upper than 30% for another individuals in elite group. Case the individual does not fit in this criteria the second best solution are analyzed and so on until the one of them satisfied this criteria. The fifth element are chosen by randomly selection of an individual of a set of the best ten individuals produced over past ten generations. The diversity index applied on this method are generated by sum of different gens in same position of two individuals compared, divided by the number of positions in the individual.

3 Computational experiments

The proposed algorithm was developed in C++ language, using Borland C++ Builder 5.0 compiler. The parameters used are obtained experimentally and they are presented in Table 1. Two instances are used to test each one of the three variants of AGA. The first one was generated by a randomly pseudo method based on works from [2], [8] and [9] with jobs numbers equal to 6, 7, 8, 9, 10, 11, 12, 15, 20, 25, 30, 35, 40, 50 and 75. The second one are generated by [11] with jobs numbers equal to 6, 7, 8, 9, 10, 11, 12, 15, 20, 25, 30, 35, 40, 50, 75 and 100. This instances are available to download in http://www.decom.ufop.br/prof/marcone/projects/scheduling/instances.htm

All of experiments were realized in a Pentium Core 2 Duo 2.1 GHz computer with 4 Gb RAM and Windows Vista operational system. Three sets of experiments are realized. The description, details and results of each one of this experiments are described in the following sections.

Table 2. Results of the first set of experiments - BAT 1

# Jobs	*Deviation* of avg			*Deviation* of best solution			Time (s)		
	AGA 1	AGA 2	AGA 3	AGA 1	AGA 2	AGA 3	AGA 1	AGA 2	AGA 3
8	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,94	0,93	0,70
9	0,15%	0,16%	0,24%	0,00%	0,00%	0,00%	1,26	1,25	0,78
10	0,24%	0,25%	0,41%	0,00%	0,00%	0,00%	1,6	1,59	0,99
11	0,03%	0,05%	0,10%	0,00%	0,00%	0,00%	2,21	2,20	1,64
12	0,07%	0,08%	0,21%	0,00%	0,00%	0,00%	2,81	2,80	2,44
15	0,76%	0,80%	1,16%	0,00%	0,00%	0,00%	6,02	5,98	6,09
20	0,73%	0,75%	0,85%	0,00%	0,00%	0,00%	20,6	20,47	17,87
25	1,02%	1,08%	1,42%	0,00%	0,00%	0,00%	45,72	45,44	39,65
30	1,60%	1,82%	2,64%	0,00%	0,00%	0,00%	112,06	111,38	41,65
40	2,33%	2,34%	3,56%	0,08%	-0,08%	-0,09%	335,88	333,81	41,61
50	4,06%	4,37%	6,32%	0,02%	-0,31%	-1,11%	896,1	890,60	222,04
75	6,52%	9,48%	11,86%	0,04%	-4,40%	-1,76%	2000,05	1992,73	1242,06
Average	**1,46%**	**1,77%**	**2,40%**	**0,011%**	**-0,399%**	**-0,246%**	**285,85**	**284,10**	**134,79**

Table 3. Comparing AGA 1 × GTPRS, proposed by [12]

# Jobs	*Deviation* of average			*Deviation* of best solution			Time (s)	
	AGA 1	GTPRS	% Improv.	AGA 1	GTPRS	% Improv.	AGA 1	GTPRS
8	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,94	0,06
9	0,15%	0,00%	-15,00%	0,00%	0,00%	0,00%	1,26	0,09
10	0,24%	0,00%	-24,00%	0,00%	0,00%	0,00%	1,60	0,15
11	0,03%	0,00%	-3,00%	0,00%	0,00%	0,00%	2,21	0,25
12	0,07%	0,00%	-7,00%	0,00%	0,00%	0,00%	2,81	0,36
15	0,76%	1,25%	64,08%	0,00%	0,00%	0,00%	6,02	0,46
20	0,73%	1,11%	51,87%	0,00%	0,00%	0,00%	20,60	2,05
25	1,02%	1,60%	56,53%	0,00%	0,00%	0,00%	45,72	6,62
30	1,60%	2,57%	60,61%	0,00%	0,00%	0,00%	112,06	18,66
40	2,33%	3,77%	61,84%	0,08%	0,00%	8,00%	335,88	84,16
50	4,06%	5,58%	37,64%	0,02%	0,00%	2,00%	896,10	305,28
75	6,52%	7,41%	13,69%	0,04%	0,00%	4,00%	2005,05	3.472,26
Avg	**1,46%**	**2,01%**	**2,27%**	**0,011%**	**-0,40%**	**-0,25%**	**285,85**	**324,20**

The first set of experiments - BAT 1 The first set of experiments uses the first instance of problems. Each set of problems are tested 30 times for AGA1, AGA2 and AGA3, variants of AGA method. Table 2 shows the results reached in the first set of experiments. The first column shows the number of jobs; the second, third and fourth column show how much the average of solutions of each variant diverted of the best solution known. In the fifth, sixth and seventh column are showed how much the best solutions generated diverted of the best solution known. In the eighth, ninth and tenth column the average of computational time of variant of AGA are showed. The results of AGA1 are compared in Table 3 with the best results founded in library, reached for [12].

The second set of experiments - BAT 2 The second set of experiments uses the second instance of problems. Each set of problems are tested 30 times

Table 4. Results of the second set of experiments - BAT 2

#	*Deviation* of average			*Deviation* of the best			Time (s)		
Jobs	AGA 1	AGA 2	AGA 3	AGA 1	AGA 2	AGA 3	AGA 1	AGA 2	AGA 3
6	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,76	0,61	0,65
7	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,76	0,94	0,81
8	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	1,03	1,21	1,00
9	0,00%	0,13%	0,20%	0,00%	0,13%	0,20%	1,35	1,53	1,26
10	0,00%	0,03%	0,06%	0,00%	0,03%	0,06%	1,61	1,88	1,65
11	0,00%	0,11%	0,17%	5,97%	0,11%	6,16%	2,20	0,73	1,92
12	0,00%	0,01%	0,11%	0,00%	0,01%	0,11%	3,32	3,09	2,37
15	0,01%	0,14%	0,37%	0,01%	0,14%	0,37%	7,49	5,60	4,38
20	0,30%	0,64%	0,81%	0,30%	0,64%	0,81%	23,88	10,18	10,61
30	1,20%	1,19%	1,35%	1,36%	1,51%	1,50%	172,51	64,19	47,13
40	0,98%	1,21%	0,98%	1,03%	1,42%	2,34%	801,67	155,05	98,05
50	1,14%	1,46%	1,14%	1,26%	2,24%	2,58%	1575,11	529,11	416,78
75	0,00%	1,26%	2,36%	0,09%	1,67%	3,60%	4978,02	3381,58	1398,82
100	0,25%	2,50%	1,10%	0,25%	3,14%	2,41%	18107,72	23209,42	15853,97
Average	**0,28%**	**0,62%**	**0,62%**	**0,73%**	**0,79%**	**1,44%**	**1834,10**	**1954,82**	**1274,24**

for each variant of AGA method, AGA1, AGA2 and AGA3. Table 4 shows the results reached in the first set of experiments. The first column shows the number of jobs. The second, third and fourth column show how much the average of solutions of each variant diverted of the best solution known. In the fifth, sixth and seventh column are showed how much the best solutions generated diverted of the best solution known. In the eighth, ninth and tenth column the average of computational time of variant of AGA are showed.

4 Conclusions

This paper treated the single machine scheduling problem with earliness and tardiness penalties, considering distinct due windows and sequence-dependent setup time. To solve this problem an adaptive genetic algorithm was proposed, where the initial population was generated by a procedure GRASP, using as a guide function dispatch rules. During the evaluation process, population pass through selection, crossover and mutation process. In crossover, five operators are used, being that the best solutions produced by each operator are submitted to local search and path relinking. The path relinking procedure connect the best solution produced to each best solutions produced by each operator.

By the end, two instances are used to test the algorithm proposed, and three variants of AGA are develop. The results of each instance are compared with another algorithm from the literature. In this experiments, the proposed algorithm presents high quality solutions with lower GAP, always reaching the best known value. The algorithm developed presents solutions better than the best solutions found in the literature, beyond presents a minor variability of final solutions, showing robustness.

Acknowledgment

The authors would like to thank CEFET/MG, CAPES and FAPERJ for the support to development of this work.

References

1. A. Allahverdi, C. Ng, T. C. E. Cheng, and M. Y. Kovalyov, A survey of scheduling problems with setup times or costs, European Journal of Operational Research, vol. 187, pp. 9851032, 2008.
2. G. Wan and B. P. C. Yen, Tabu search for single machine scheduling with distinct due windows and weighted earliness/tardiness penalties, European Journal of Operational Research, vol. 142, pp. 271281, 2002.
3. J. Du and J. Y. T. Leung, Minimizing total tardiness on one machine is np-hard, Mathematics of Operations Research, vol. 15, pp. 483495, 1990.
4. C. Y. Lee and J. Y. Choi, A genetic algorithm for job sequencing problems with distinct due dates and general early-tardy penalty weights, Computers and Operations Research, vol. 22, pp. 857869, 1995.
5. T. A. Feo and M. G. C. Resende, Greedy randomized adaptive search procedures, Journal of Global Optimization, vol. 6, pp. 109133, 1995.
6. A. C. Gomes Jr., C. R. V. Carvalho, P. L. A. Munhoz, and M. J. F. Souza, Um método heurístico híbrido para a resolução do problema de sequenciamento em uma máquina com penalidades por antecipação e atraso da produção, in Anais do XXXIX Simpósio Brasileiro de Pesquisa Operacional XXXIX SBPO, Fortaleza, Brazil, 2007, pp. 16491660.
7. M. Prais and C. C. Ribeiro, An application to a matrix decomposition problem in tdma traffic assignment, INFORMS - Journal on Computing, vol. 12, pp. 164176, 2000.
8. C. F. Liaw, A branch-and-bound algorithm for the single machine earliness and tardiness scheduling problem, Computers and Operations Research, vol. 26, pp. 679693, 1999.
9. R. Mazzini and V. A. Armentano, A heuristic for single machine scheduling with early and tardy costs, European Journal of Operational Research, vol. 128, pp. 129146, 2001.
10. G. Rabadi, M. Mollaghasemi, and G. C. Anagnostopoulos, A branch-and-bound algorithm for the early/tardy machine scheduling problem with a common due-date and sequence-dependent setup time, Computers and Operations Research, vol. 31, pp. 17271751, 2004.
11. B. F. Rosa, M. J. F. Souza, and S. R. Souza, Formulações de programação matemática para o problema de sequenciamento em uma máquina com janelas de entrega distintas e tempo de preparação dependentes da sequência de produção, in Anais do XXXII Congresso Nacional de Matematica Aplicada e Computacional CNMAC 2009, Cuiaba, 2009.
12. P. H. V. Penna, Um algoritmo heurístico híbrido para minimizar os custos com a antecipação e o atraso da produção em ambientes com janelas de entrega e tempos de preparação dependentes da sequência de produção, Dissertação de mestrado, Programa de Pós-Graduação em Engenharia Mineral, Escola de Minas, UFOP, Ouro Preto, 2009.